NUMERICAL CONTROLLER FOR MILLING MACHINE

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

By
V. MANUJA

to the
DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
AUGUST, 1976

- [illegible] JA[illegible]

EE-1976-M-MAN-NUM

TO MY PARENTS

CERTIFICATE

This is to certify that the thesis entitled, 'Numerical Controller for Milling Machine' describes the work carried out by Mr. V. Manuja under our supervision and that it has not been submitted elsewhere for a degree.

A. Ghosh
Professor
Dept. of Mech. Engg.
I.I.T. Kanpur

R. N. Biswas
Assistant Professor
Dept. of Elect. Engg.
I.I.T. Kanpur

August 1976.

ABSTRACT

A numerical controller has been designed and fabricated for a milling machine providing continuous two-dimensional position control in the horizontal plane with the facility to adjust the vertical position manually. The machine table is driven by stepper motors which are digitally controlled by the instructions received from an input program. Each instruction corresponds to data for an interpolation interval or an OP code. The data for each interpolation interval consists of three words, each of 12-bits (i) the length of cut 'L' (ii) its x-component (X) and (iii) its y-component (Y), which are fed to the system serially. PRM technique has been used to implement the linear interpolation principle.

The system provides various operational facilities viz. floating zero, variable feed drive, fast return to origin and safety measures to protect the machine. The controller provides various displays on the console depicting its status at all the times. It is automatic in operation except a few initial settings.

A two-dimensional bed-positioner, simulating a milling machine work-piece table, has been fabricated for demonstration. The system has been tested and found to be satisfactory.

ACKNOWLEDGEMENT

I am very much indebted to Dr. R.N.Biswas for suggesting an interesting topic and for the excellent guidance rendered throughout this work. Words do not suffice to convey my deep sense of gratitude towards him for the whole lot of experience I gained in course of this work.

I am highly grateful to Dr. A. Ghosh for many valuable suggestions and in the design and fabrication of the mechanical set-up without which the factual demonstration would not have been possible.

I express my sincere thanks to Mr. T.P.Verma for his constructive criticism and useful discussions. My thanks are due to Mr. R.M.Jha and Mr. B.P.Vishwakarma of TA 203 Lab., who played a major role in the fabrication of mechanical set-up. I am also thankful to E.E.Department and ACES personnel, especially Mr. Sikdar, who co-operated at every stage of the work.

I am obliged to my friend Mr. Rajgopalan for his help during the preparation of the manuscripts.

Finally, I thank all those who have helped me directly or indirectly in bringing the project to a success.

LIST OF FIGURES

FIG.NO.	CAPTION	Page
2.1	4-Bit PRM	8
2.2	Error in linear interpolation	10
3.1	Plate supporting scheme	15
3.2	M.S. Wedge	15
3.3	Supporting Block	15
3.4	Load screw nut	17
3.5	Mechanical set-up	18a
4.1	Block diagram	20
4.2	Display logic	24
4.3	Logic circuit for display	25
4.4	Sensor and error-detection unit	27
5.1	Flow chart	33
5.2	Demand pulse generator	36
5.3	Interrupt circuitry	38
5.4	Timing Diagram	40
5.5	Code_detection and storage circuit	42
5.6	Gating circuit	44
5.7	Indication circuit	45
5.8	Feed clock generator	47
6.1	Principle of stepper motor	49
6.2	Sequence Generator	51
6.3	Logic circuit	53
6.4	Single pulse and clock-burst generator	55

6.5	Power stage	55
7.1	Buffer-divider-decoder	58
7.2	Buffer-PRM-counter	59
7.3	Control-I	60
7.4	Control-II	61
7.5	Control-III	62
7.6	Translator	63
7.7	Position display	64
7.8	Passes and Cycles Display	65
8.1	Test Circuit	71a
8.2	Test Pattern	72b

LIST OF TABLES

TABLE NO.		Page
5.1	OP Codes	30
5.2	Interrupt Facilities	31
7.1	Functional Description of Cards	57
7.2	Card Layouts	66
7.3	Card Pin-connections	66a
7.4	Trouble Shooting Chart	70a
8.1	Data Input by Test-Box	72
8.2	Test Pattern Table	72a

LIST OF SYMBOLS

EREG	-	Buffer-Register
C_D	-	Mode of display counter
C_K	-	Load pulse
C_o	-	Free running clock
CREG	-	Code-register
DIR INT	-	Motor direction in manual mode
DS	-	Disable start
D_{xy}	-	Decoder output of 'x' and 'y' counter
EREG	-	Execution-register
EXT DIR	-	Motor direction in auto mode
$\bar{H}$	-	Magnetic field
IE	-	Input error
I_{FF}	-	Interpolation flip-flop
L	-	Length of interpolation interval
M	-	Mode
M_x/M_y	-	Output state of x/y counters
NC	-	No connection
OE	-	Output error
P	-	Logical variable for position
$\bar{Q}_R$	-	$\bar{Q}$ of Ready latch
$\bar{Q}_S$	-	$\bar{Q}$ of Stop latch
Q_{SFF}	-	Q of sensor flip-flop
$\bar{Q}_{SYNC}$	-	$\bar{Q}$ of Sync Latch
R	-	Clear
S_d	-	Sign of input data
S_D	-	Sign of display

TP	-	Test point
V_{cc}	-	Power supply for IC's
V_d	-	Power supply for display
V_M	-	Power supply for stepper motors
x/y	-	x/y component of length
τ	-	Delay

CONTENTS

		Page
CHAPTER 1	INTRODUCTION	1
CHAPTER 2	NUMERICAL CONTROL HARDWARE	4
2.1	Digital Drive	4
2.2	Linear Interpolation	5
2.3	Pulse Rate Multiplier	7
2.4	Error in Linear Interpolation	9
CHAPTER 3	MECHANICAL SYSTEM	12
3.1	System Specifications	12
3.2	Supporting Block and Table Plates	14
3.3	Lead Screw Nut	16
3.4	Reduction Gear	16
3.5	System Description	18
CHAPTER 4	OVERALL SYSTEM DESIGN	19
4.1	Operating Features	19
4.2	Serial Access Memory	21
4.3	Programmable Divider and Decoder	22
4.4	Register and Pulse Rate Multiplier	22
4.5	Down Counters	22
4.6	Logic for Display	23
4.7	Counters and Display	26
4.8	Sensor and Error Detection Unit	26
4.9	Control Unit and Translator	26

CHAPTER 5	CONTROL UNIT	28
5.1	Control Unit Flow-Chart	29
5.2	Demand Pulse Generator	35
5.3	Interrupt Circuitry and Sync.Latch	37
5.4	Code Detector and Code Register	39
5.5	Gating Circuit	41
5.6	Indication Circuit	43
5.7	Feed Clock Generator	46
CHAPTER 6	STEPPER MOTOR TRANSLATOR	48
6.1	Stepper Motor	48
6.2	Sequence Generator	50
6.3	Control Logic	52
6.4	Single-Pulse and Clock-Burst Generator	54
6.5	Power-Stage	54
CHAPTER 7	LAYOUT AND OPERATING INSTRUCTIONS	56
7.1	Circuit Layouts	56
7.2	Power Switches	56
7.3	Panel Displays	56
7.4	Panel-Control Switches	67
7.5	Panel Connectors	68
7.6	Operating Instructions	68
CHAPTER 8	TEST RESULTS AND CONCLUDING REMARKS	71
BIBLIOGRAPHY		73
APPENDIX		75

CHAPTER 1

INTRODUCTION

Automation plays a vital role in the industrial world of today. In almost every field a greater amount of work is being done by machines, with the result that the human skill involved is a minimum. Numerical control of machines is one of the key features of automation. In the present project, an attempt is being made to evolve a scheme by which numerical control can be incorporated in an ordinary milling machine, simply by adding a digitally driven job positioner and a programmed controller for the digital drive.

Numerical control implies, process control by numerals, i.e. it utilizes digital circuits to cause the system work in accordance with instructions coded in numbers, given to the system through an interface. It differs from the analogue since it utilizes only the presence or absence of signals to operate. The basic difference between the human and numerical control lies in the former's ability to think originally. Numerical control replaces human operators, but can do only those jobs for which the designer has made it capable and nothing more.

There are various types of numerical control, depending on the job to be performed. It has been widely used for the machine tools e.g. finite position control

system; 'point to point' control for processes as boring, drilling; two axis continuous positioning, etc.

Numerical control has a wide scope for usage. Different systems are built for different jobs to be performed and hence the cost of the system depends on the complexity to be introduced. With the advent of microprocessors, it is possible to design extremely versatile numerical control system with the flexibility limited only by the software. However, most practical N/C systems needed in machine shop operations like the problem under our consideration have essentially to be custom-made for specific types of jobs because the cost of a precision machine is usually much more than that of the controller. The present thesis is therefore restricted to the design of hardware based on the standard available digital IC's (TTL 7400 series). This, as borne out by the actual design, is capable of sufficient flexibility with a simple low cost control system.

The next chapter is devoted to the principles of the hardware for the numerically controlled milling and explains its advantages and limitations. The construction of the mechanical model simulating a milling machine is taken up in Chapter 3.

Chapter 4 deals with the overall system design. The block diagram is explained with some of the blocks

explained in detail. The CONTROL unit has been explained in Chapter 5, starting with a flow-chart bringing out the logical organisation of the system, followed by explaination of its hardware implementation.

The design of the translator for the stepping motor is presented in Chapter 6 followed by the system layout and the operating instructions in Chapter 7.

The thesis is concluded with a chapter on the test procedure and results obtained.

CHAPTER 2

NUMERICAL CONTROL HARDWARE

The principles of the proposed hardware for numerically controlled milling are discussed in this chapter. Section 2.1 describes the advantages of digital drive, Sections 2.2 to 2.4 explain the linear interpolation [1] and the error that is incurred, and also brings into focus the pulse rate multiplier.

2.1 DIGITAL DRIVE

A system is said to be numerically driven if it utilizes digital hardware to control mechanical movements with digitally coded instructions. The drive should be digital in nature to be able to control the feed rate and to maintain at the same time the desired precision, i.e. the rotation and the speed of the motors have to be directly determined by the number of drive pulses and the rate at which these pulses are applied. One possible method to achieve this is to use a closed-loop D-C motor drive through a digital-to-analogue interface. Apart from the added cost and complexity, this obviously restricts the accuracy by the digital-to-analogue converter. A stepper motor is ideal for this, since it is inherently capable of giving an error-free correspondence between the number of applied pulses and the number of steps of

its rotation. A stepper motor, however, lacks smoothness in rotation because its rotation is basically in discrete steps and as such, one can only obtain an average feed velocity of the required value.

2.2 LINEAR INTERPOLATION

In general a trajectory is specified numerically either in the form of a series of successive values of coordinates or a series of successive increments along the coordinate axes. The latter choice is preferable since it being smaller values, it can be handled with greater ease. The successive increments are so chosen that a particular geometric curve joining the end points does not differ from the actual trajectory by more than the tolerance specified. Depending on the choice of the curve one may obtain linear, circular or parabolic interpolation. We shall, however, restrict our attention to linear interpolation where the programme breaks up the trajectory into small linear segments.

Let

'L' be the desired length of cut

'X' be the desired displacement in x-direction

'Y' be the desired displacement in y-direction.

Now if V_f be the required feed velocity, then the required velocities along the x and y axes to produce V_f are given by

$$V_x = \frac{X}{L} V_f \tag{2.1}$$

$$V_y = \frac{Y}{L} V_f \tag{2.2}$$

As the velocities are proportional to the corresponding pulse rates, we can write

$$V_x^* = V_f^* \frac{X}{L} \tag{2.3}$$

and $$V_y^* = V_f^* \frac{Y}{L} \tag{2.4}$$

where * indicates pulse rates.

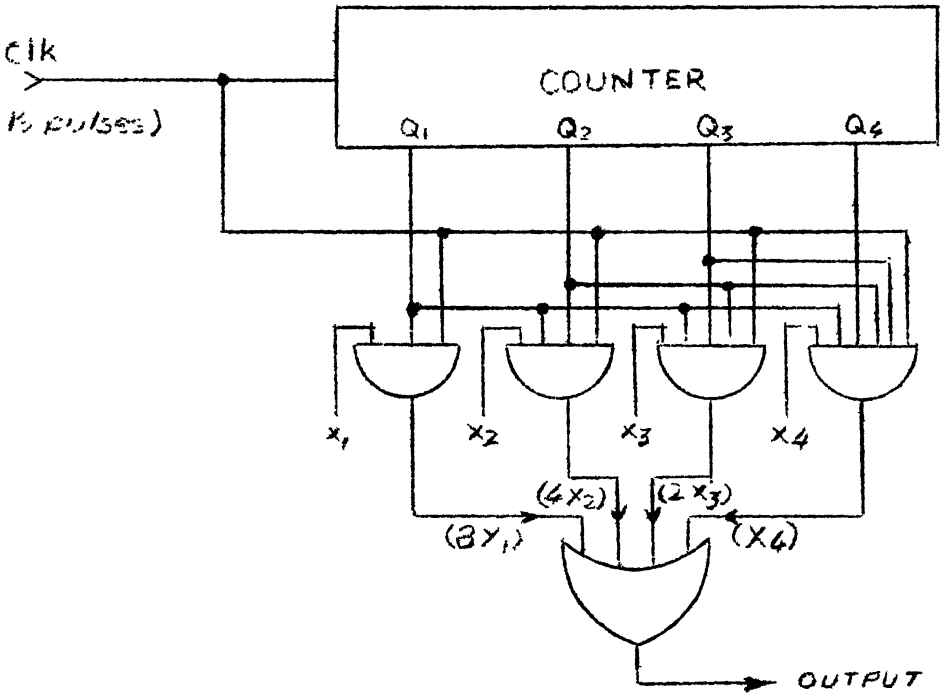
The problem of achieving linear interpolation of a given trajectory may be divided into two parts:
(i) Generation of pulse trains at rate proportional to V_x and V_y as given by equations (2.3) and (2.4), and
(ii) Sending of exact numbers of pulses at these rates proportional to the numbers x and y, to the x and y drives respectively. The second part of the problem is relatively simple to solve, one of the common solutions being presettable down counters. The generation of appropriate pulse rates, however, is a more involved problem. There are two possible solutions, but none of them is ideal. One possibility is to use a phase-locked loop (PLL) which can, in principle, give an output frequency proportional to any control data. However, since the frequency range required by equations (2.3)

and (2.4) is rather large, capture problems associated with a PLL are likely to be quite difficult. The other alternative is a pulse rate multiplier (PRM), which can generate output pulses at an average rate proportional to a control data. However, the output pulses generated by a PRM are not uniformly spaced and hence lead to interpolation errors, which are discussed later. The PRM has been chosen in preference to the PLL because of its easy compatibility with digital drive. The principle of a PRM has been taken up in the next section.

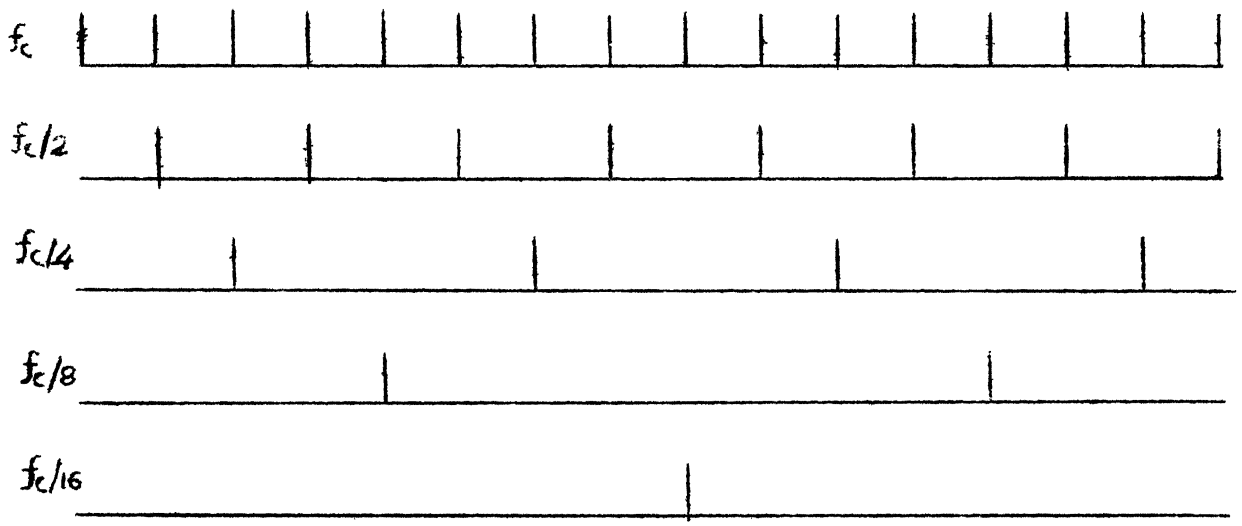
2.3 PULSE RATE MULTIPLIER

A PRM consists of a counter and a set of decoders which gate a continuous train of input pulses to generate the output pulses in accordance with a control input. The input pulse train is usually known as the rate input, and the control input, as the data input having its bits fed in parallel. The number of output pulses in one cycle of the counter is given by the magnitude of the data input provided that the data and the counter are compatible with each other.

Figure 2.1 illustrates a four-bit PRM with parallel data fed in binary representation, the decoders being simple AND gates. The number of pulses passing through the decoder of one cycle have been marked in parenthesis on the figure. The pulses coming out of one decoder never



(a) CIRCUIT



(b) TIMING DIAGRAM

FIG-2.1 4-BIT PRM

coincides with the pulses coming out of any other decoder. This property is the key to the operation of pulse rate multipliers, for it allows the pulses coming out of the different decoders to be combined by a simple OR gate to obtain an output having a sequency given by a simple algebraic sum of the pulses at the inputs:

$$S_o = f_c \sum_{i=1}^{n} \frac{X_i}{2^i} \qquad (2.5)$$

$$f_c = \frac{X}{2^n} \qquad (2.6)$$

where S_o is the output sequency,

f_c is the frequency of the rate input

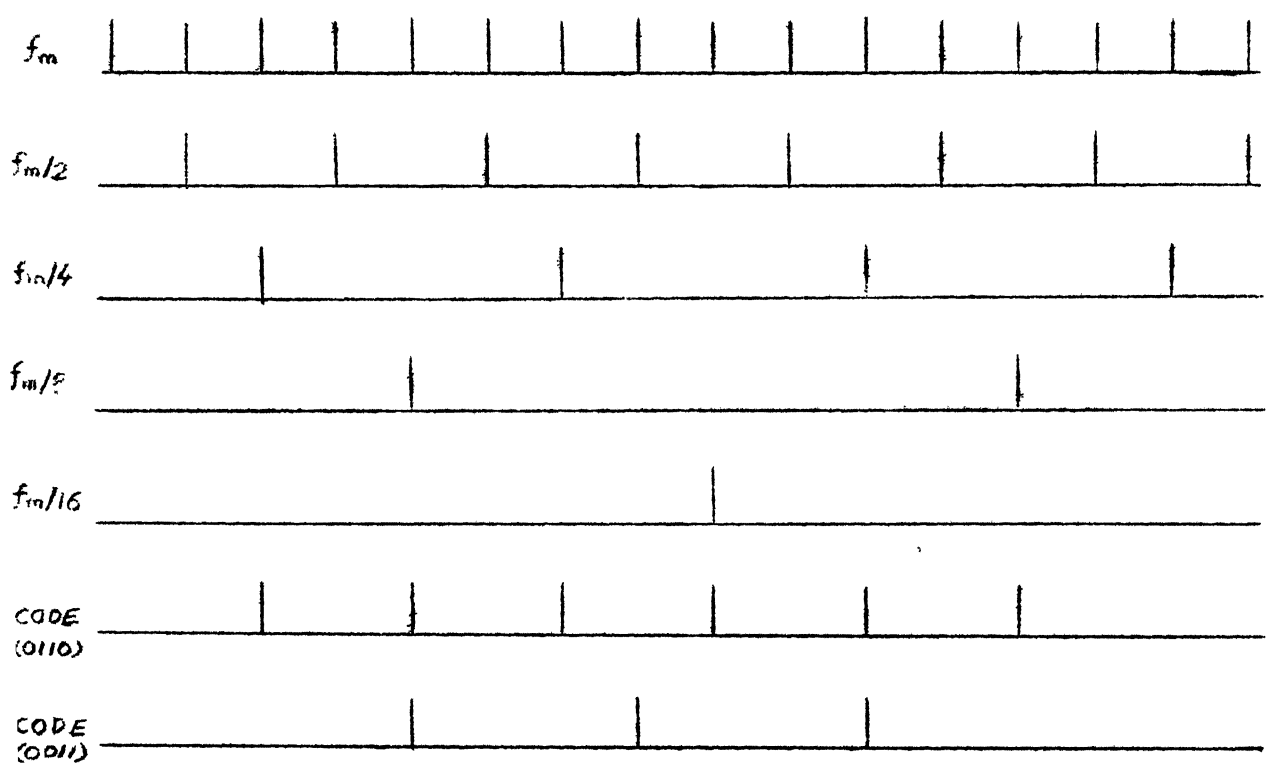
X is the data input

and $X_1, X_2, \ldots, X_n$ are the bits representing X.

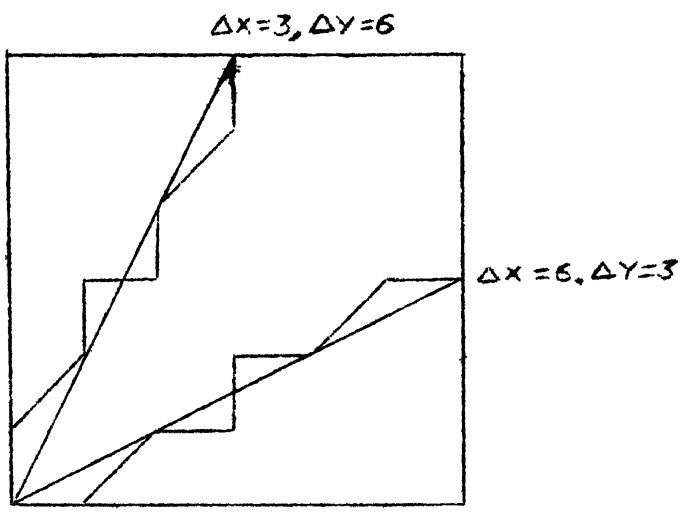
2.4 ERROR IN LINEAR INTERPOLATION

It has been pointed out in Section 2.2 that a PRM gives output pulses at a non-uniform rate. As the drive motors are driven digitally the actual movement can have only three slopes. The slope is 0°, 45° or 90° depending on whether only an 'x' pulse comes, an 'x' pulse and a 'y' pulse occur simultaneously, or only a 'y' pulse comes, respectively. This gives rise to an error in linear interpolation and has been illustrated in Figure 2.2.

It is clear from the example given in Figure 2.2 that by virtue of the discrete as well as non-uniformly



(a) WAVEFORM



(b) ERROR ILLUSTRATION

FIG-2.2 ERROR IN LINEAR INTERPOLATION

distributed nature of the drive pulses, the output motion cannot exactly follow the straight line whose slope is given by y/x. As has been shown in Figure 2.2, the trajectory of motion is given by a zig-zag line which is symmetrically oriented with respect to the desired straight line. The zig-zag line joins the same two end points of the spatial interval. A general expression has been derived by Karibskii [2] for the magnitude of the deviation 'Δ' of the zig-zag line from the required line. Karibskii estimated the absolute upper limit for 'Δ' to be $n/\sqrt{2}$ where n is the maximum number of data bits.

CHAPTER 3

MECHANICAL SYSTEM

To test the performance of the circuitry developed for numerical control of milling machine, a mechanical system has been fabricated. This, model set-up simulating milling machine work-piece table is necessitated because of the non-availability of stepper motors with high torque and/or high r.p.m. in India to drive the conventional milling machine. Brief description of this model set-up has been presented in this chapter.

3.1 SYSTEM SPECIFICATIONS

The system, as already mentioned, has been built for demonstration of the Digital control. To find out the accuracy that can be achieved by the system, it is necessary to mention the details of the data that can be fed to the Digital Control. The minimum length of interpolation is taken to be 0.1 mm. It is based on the assumption that any engineering curve can be linearly interpolated, with the minimum length of interpolation being 0.1 m.m. The least count of the system has been assumed to be 0.01 mm., i.e. for every pulse to fed motor, the distance covered is 0.01 mm. These are fixed by the number of bits for a data that have been provided in the system. It has been pointed out in Chapter 2 that for every interpolation interval, three things are to be given,

viz., (1) length, (2) component in x-direction and (3) component in y-direction. The data length is taken to be 12 bits, out of which one bit is required for parity in each case and another is required for sign of the data for x and y components. Hence each component can have a maximum magnitude of 10 bits. As the least count of the system is 0.01 mm., the maximum length that the 10 bits can represent is $(2^{11}-1)$x 0.01 $\simeq$ 20 mm., i.e. maximum component on either axis can be 2 cms in any interpolation interval. Accuracy that should be obtained is 0.01 mm which is the least count of the system, but is not achievable because of the inherent error in linear interpolation which has been described in detail in Section 2.4.

Hence accuracy desired is 0.01 mm.

The total travel in the forward and cross directions has been chosen to be smaller than what is provided in an actual milling machine for the sake of portability of the simulated bed. However, care has been taken to ensure that contours of reasonable size can be demonstrated. The chosen values are:

Total travel in the forward direction = 20 cms

Total travel in the cross direction = 15 cms.

For the desired accuracy the lead screw pitch required with this system is 0.01x200 = 2 mms. The lead screw of this pitch was not available, but a lead screw with a pitch of 6.3 mm. (1/4"), which was available, has been used.

This model should be capable of giving continuous two-dimensional movement when driven by the torque available from the stepper motor, which is much smaller than that required for driving a conventional milling machine bed, This is because of the sliding friction due to its dove-tail construction. Hence a different type of arrangement utilizing roller bearings has been used here since rolling friction is much less than sliding friction.

3.2 SUPPORTING BLOCK AND TABLE PLATES

It has been mentioned in the previous section that the set-up should have low-friction to allow free movement with smaller torque. The supporting blocks have to meet two requirements

(1) To prohibit any movement in the vertical direction.

(2) To stop any movement in the lateral direction.

This can be done by supporting the plates at 45° as shown in Figure 3.1. The aluminium plates have been chosen for light-weight consideration and hence M.S.Wedges have been fixed at the ends of the plates to avoid fast wear and tear (Figure 3.2). The supporting blocks (Figure 3.3) have also been made of aluminium. The number of such blocks has been so chosen that the load shared by each bearing does not exceed its limit.

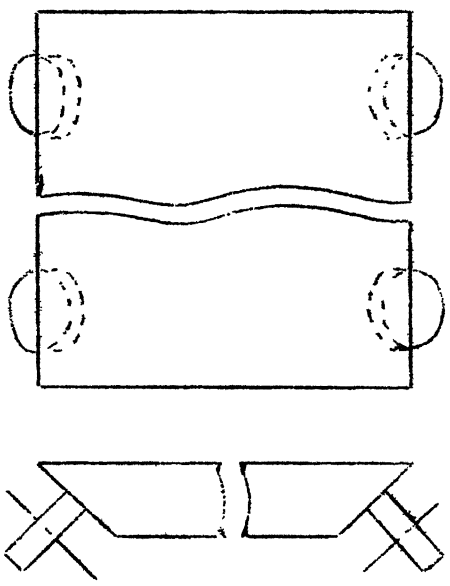


FIG-3.1 PLATE SUPPORTING SCHEME

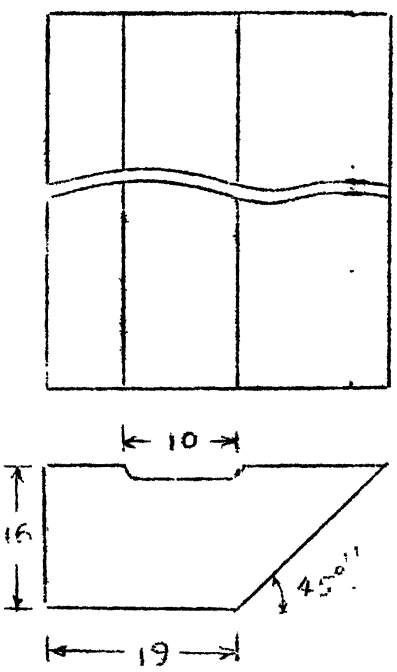


FIG-3.2 M.S. WEDGE

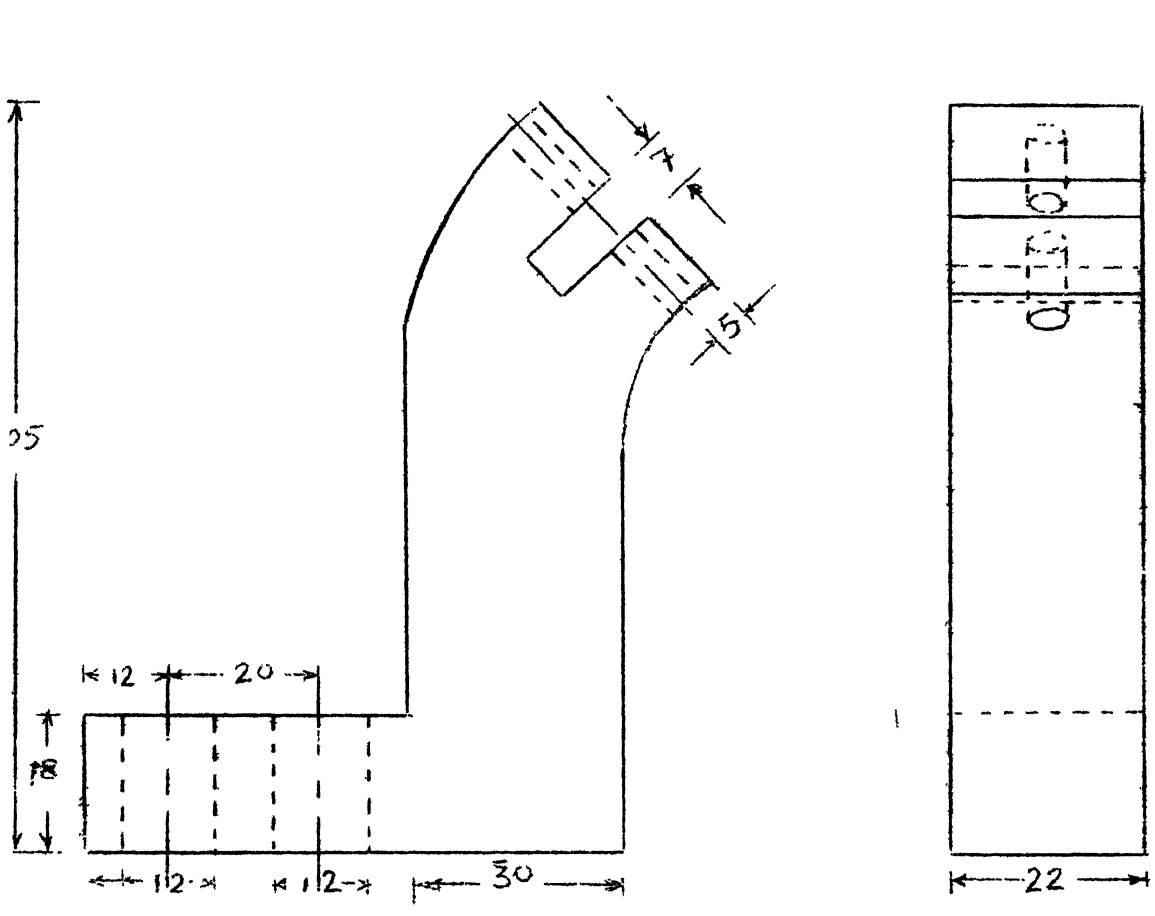


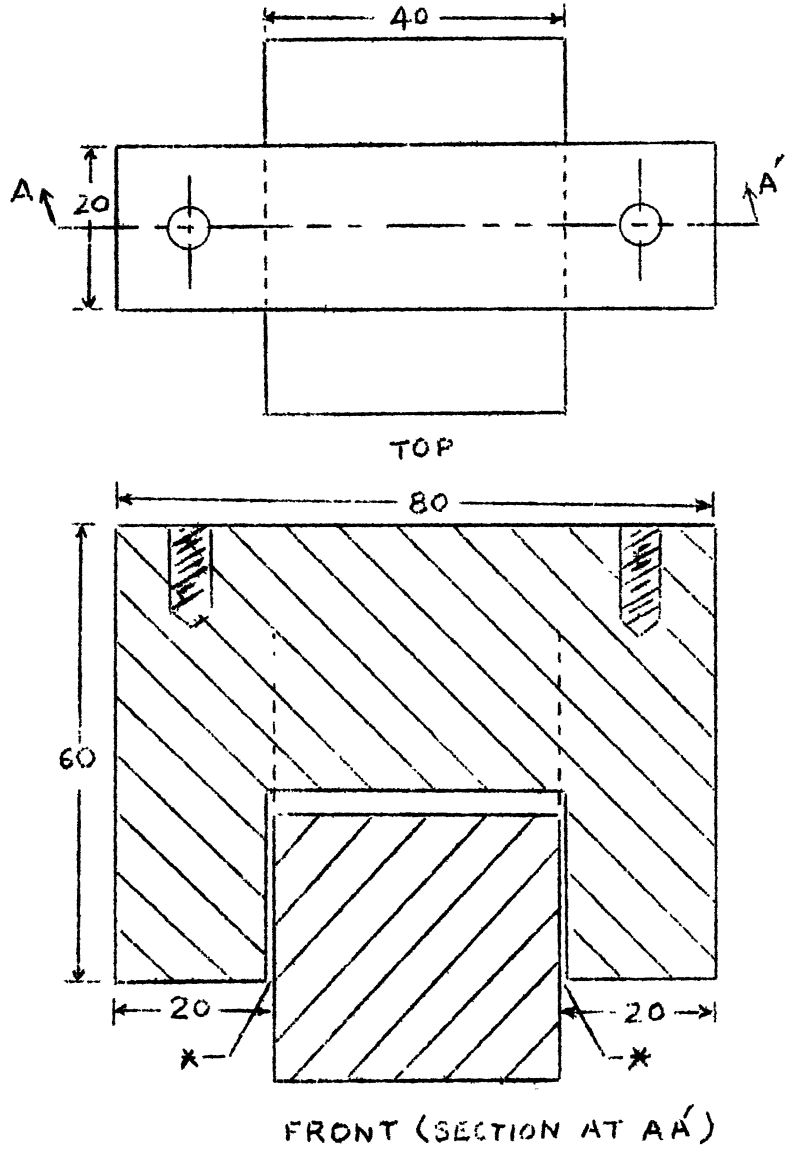
FIG-3 3 SUPPORTING BLOCK

3.3 LEAD SCREW NUT

The lead screw nut is designed so that small inaccuracies in the straightness of the lead screw do not require additional force for driving the machine table. To accomplish this, the lead screw nut is made in two parts, one sliding in the other (Figure 3.4). The surfaces which slide are accurately machined, the result being that the back-lash problem is a minimum but still permits smooth sliding. Care is taken in preparing lead-screw nut so that it matches with the lead-screw providing a minimum amount of back-lash.

3.4 REDUCTION GEAR

The stepper motor is inherently a low speed drive. Hence gears for reducing the speed are not needed. A direct coupling is used for connecting the stepper motors to the system. The stepper motor needs 200 pulses for completing one rotation, i.e. every step makes an angular movement of 1.8^{o}. For ensuring accurate, continuous movement of the system, it is essential to have a transducer to track the mechanical movement continuously. A disc with optical gratings has been used for the purpose but as the angular movement per step is 1.8^{o}, the required disc size to enable the transducer to pick up every step movement becomes very large. Hence a step-up gear is



* CLEARANCE IS .01 m.m.

FIG-3.4 LEAD SCREW NUT

used to reduce the disc size and still permitting it to pick up movement due to each and every step.

3.5 SYSTEM DESCRIPTION

The system has been fabricated in accordance with the specifications spelt out in Section 3.1. The principal components have been explained in previous sections. The M.S. Wedges and half of lead screw nut is fixed to the respective table plates. The other half of lead screw nut is put in the lead screw which is supported at the ends by conventional ball-bearings fitted in plummer blocks. The stepper motors are rigidly coupled to the respective lead screws. The complete set-up is shown in Fig.3.5.

CHAPTER 4

OVERALL SYSTEM DESIGN

This chapter deals with the various requirements of the system to meet the specifications spelt out in the previous chapter. Section 4.1 describes the requirements arising out of the features to be incorporated for the convenience of the user. The descriptions of the various blocks shown in the block diagram (Figure 4.1) are taken up in the subsequent sections.

4.1 OPERATING FEATURES

The word-size requirement has been decided in Chapter 3 to be 36 bits for each interpolation interval. Since the entire program for a particular N/C (Numerical Control) job is to be serially accessed from a serial access memory (SAM) and then executed, two registers are required: (i) An execution register (EREG) holding the data being executed, and (ii) A buffer register (BREG) holding the next data to be executed. This enables continuous operation without waiting for the access of the next data from the SAM. Besides these basic requirements the following facilities are to be provided for the convenience of the user:

1. Facility to provide a 'floating zero'.

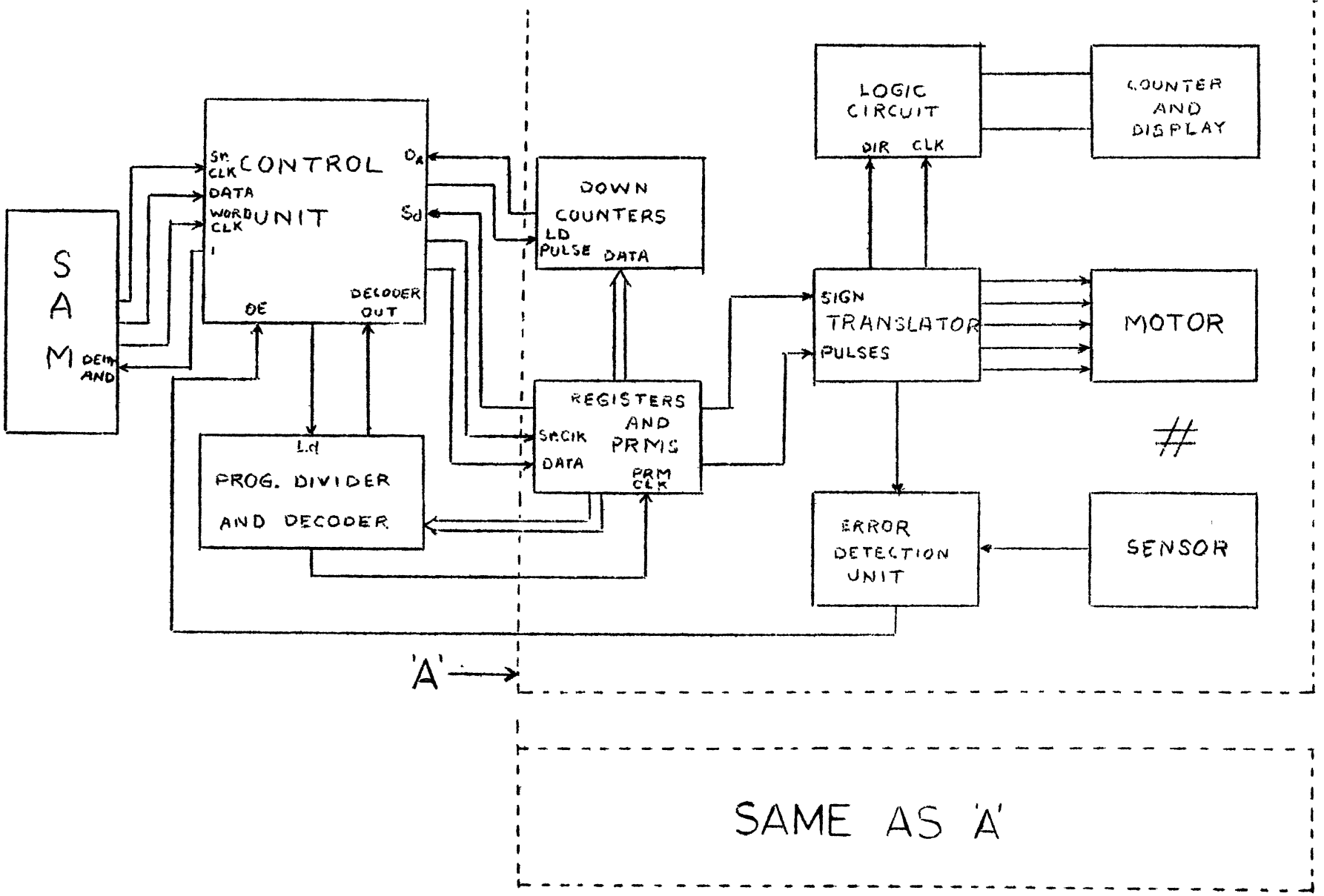


FIG-4.1 BLOCK DIAGRAM

2. Fast return to the starting point, if so desired, after the completion of a pass.
3. Automatic stop, enabling manual intervention at certain pre-programmed points, marking the ends of the cycles.
4. Automatic stop and indication when a job is completed.
5. Manual halting of the system at any point during the process without any loss of data.
6. Manual stop without any relevant data being left in the system.
7. Continuous control of feed rate.
8. Error indication in case data does not come from SAM inspite of the demand.
9. Disabling of control and error indication in case the motor movements do not correspond to the drive pulses.
10. Numeric displays of the position, number of passes and number of cycles.
11. Indication of the state of the controller (Halt, Stop, Error, Ready etc.).

4.2 SERIAL ACCESS MEMORY (SAM)

The memory used here, SAM, is being designed in another M.Tech.project, which gives one word (in this case 12 bits) of data whenever a demand pulse is sent to it. It provides the system the data, the bit clock as well as the word clock which coincides with the last bit block of the word.

4.3 PROGRAMMABLE DIVIDER AND DECODER

A programmable divider basically gives the scaling factor which was written as 'L' in equations (2.3) and (2.4), to the feed clock which is directly proportional to the feed velocity. This has been realized by using presettable up/down binary counters (SN 74191) because the data is in binary form. The decoder is used to sense the condition when all the bits in data 'L' are zero, and under this condition to inhibit the clock pulses going to the PRM.

4.4 REGISTERS AND PULSE RATE MULTIPLIER

The need of the registers has already been pointed out in Section 4.1 and the principle of a PRM has been explained in Section 2.3. This has been realized by cascading two 6-bit PRM chips (SN 7497). Two most significant bit inputs have been grounded as only 10 bit data is fed to it.

4.5 DOWN COUNTERS

The purpose of the down counters is to supply the exact number of pulses for every interpolation interval. The down counter is preset with the x- or y-data, as applicable, simultaneously, with the loading of the same data into the execution register from buffer register. After the required number of pulses have been received from the respective PRM, the down counter reads zero and sends a command to the control unit which disables the corresponding

pulse rate multiplier. This has been realized by cascading 4-bit presettable up/down counters (SN 74191).

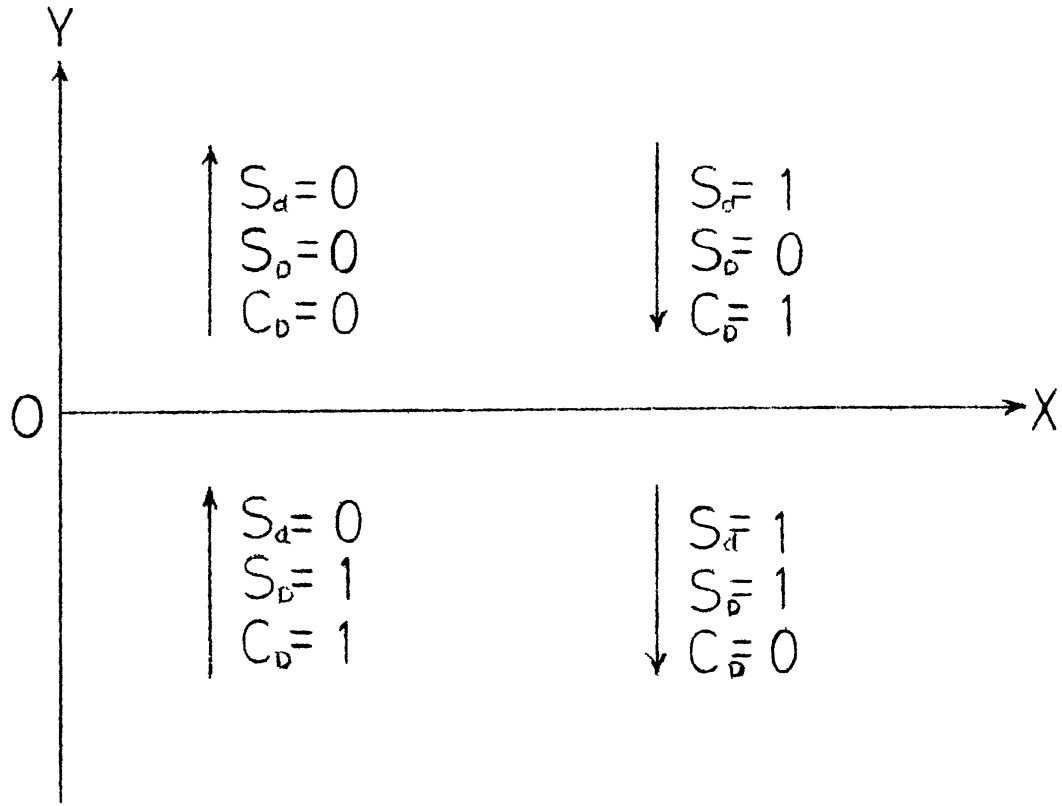
4.6 LOGIC FOR DISPLAY

A simple circuit is developed to provide FLOATING ZERO facility to the user. The requirements of the logic circuit is explained in Truth Table, shown in Figure 4.2a and illustrated in Figure 4.2b. There are two independent variables, S_d, the sign of data input and P, the logical position variable of the machine table and it is '0' when the location is positive, i.e. above the axis and '1' when it is negative, i.e. below the axis, and, two dependent variables S_D, the display sign and C_D, the display counter mode.

It is easily seen from the Truth Table in Figure 4.2a that display sign does not change until P changes its value, i.e. it crosses the axis. This is realized by an edge triggered 'D' flip-flop at the instant when P changes and takes the value in accordance with S_d at that time. The counter mode should change whenever the sign of the fresh data obtained changes or, when display sign changes, but should not alter it both of them change at the same time. This has been realized by an EX-OR gate, its inputs being S_d and S_D. To ensure correct display, the counter mode should change before next pulse is fed to it hence a delay 'τ' is provided for the pulses going to the counter input. The complete circuit is shown in Figure 4.3.

S_d	P	S_D	C_D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

(a) TRUTH TABLE



(b) ILLUSTRATION OF TRUTH TABLE

FIG-4.2 DISPLAY LOGIC

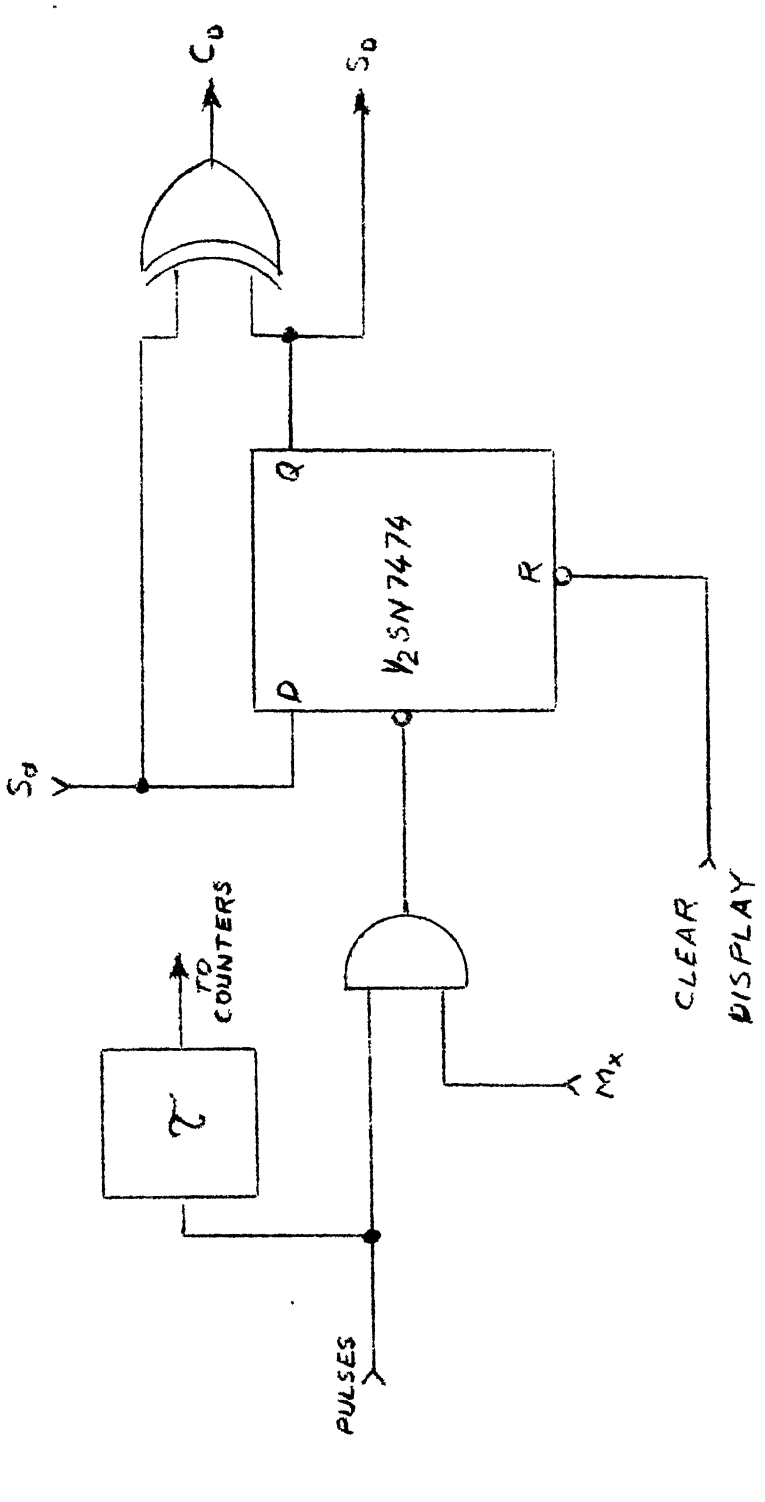


FIG-4.3 LOGIC CIRCUIT FOR DISPLAY

4.7 COUNTERS AND DISPLAY

The position of machine table at all times is displayed using 7-segment display. The counter being used to keep track of the position continuously is realized by using synchronous up/down decade counters (SN 74190). For decoding the counter state and driving 7-segment display, decoder driver (SN 7447) is used. These counters also provide look ahead carry signals called max./min. outputs, which when combined in a AND gate provide enabling input, called M_x shown in Figure 4.3, to the D-flip-flop.

4.8 SENSOR AND ERROR DETECTION UNIT

To check continuously the movement of the mechanical system a transducer, which can give the output for every step movement by the motor is needed. For this purpose a photo-transistor which gives out a pulse whenever light reaches it by the movement of a disc with optical gratings, is utilized. The circuit for this purpose is shown in Figure 4.4. Error detection unit continuously keeps track of the number of input pulses and pulses through the sensor and whenever the difference goes beyond the limit, an error signal is sent to the control unit disabling the system from sending further drive pulses and an error indication is lighted.

4.9 CONTROL UNIT AND TRANSLATOR

These have been explained in detail in Chapters 5 and 6 respectively.

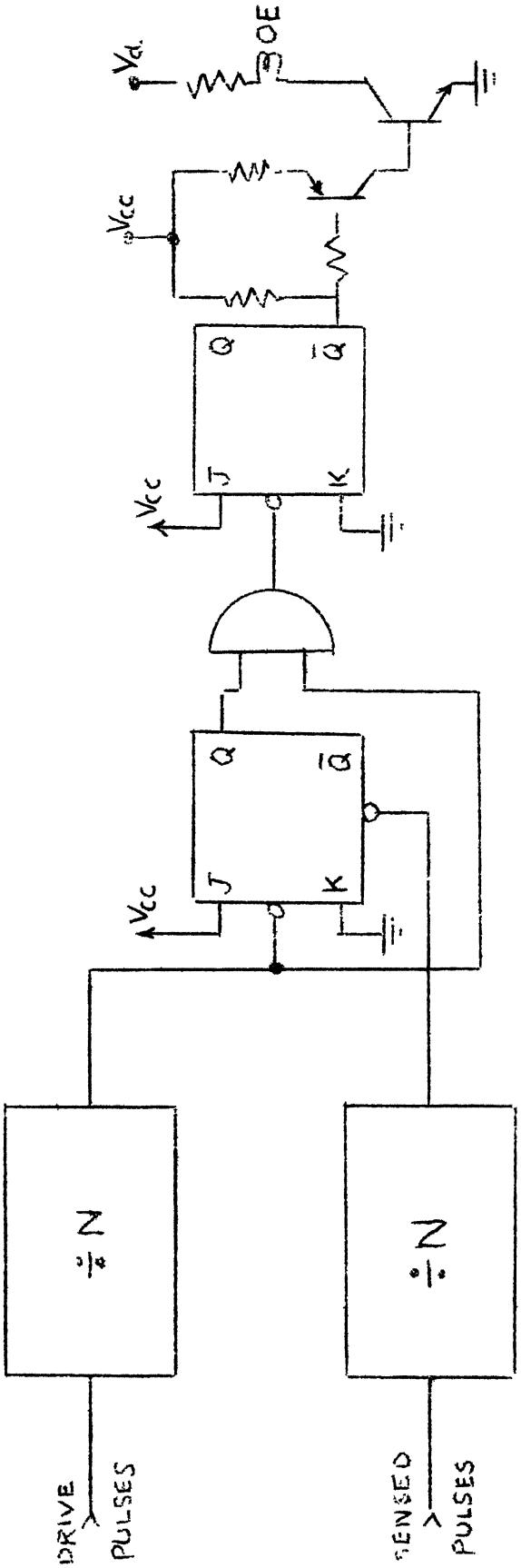


FIG-4.4 OE DETECTION CIRCUIT

CHAPTER 5

CONTROL UNIT

The control unit is responsible for the execution of the job in accordance with the commands, which may be given to it manually from panel switches or electrical from coded input data. Depending on the type of command, one can over-ride the other commands. The requirements of the control unit have already been pointed out in the previous chapter and these are put briefly in the light of the hardware required in the first section along with the flow-chart to help in understanding the control unit of the system. The circuit required for generating the demand pulses to obtain the data from the SAM has been dealt with in the next section, while the hardware required for interrupting the system manually or otherwise is described in the one next to that, followed by the code detection scheme. The provision of the fast return to the 'Defined Origin' and the indication of the completion of a cycle or the job are then discussed and the chapter is concluded with a description of the circuit for generation of the feed clock used to vary the feed velocity of the machine.

5.1 CONTROL UNIT FLOW-CHART

The various operating features chosen for the system have been mentioned in Section 4.6. These features may be broadly grouped under the following three heads:

(1) Uninterrupted Execution of Input Data:

During normal running of the program, in absence of any 'Op Code' or 'Interrupt', the values of L, X and Y relevant to the current interpolation interval are stored in EREG, while those corresponding to the next are stored in BREG. The completion of the execution of the data is marked by a pulse D_{xy}, generated by the appropriate decoding of the states of the up/down counters. Each Dxy pulse therefore transfers data from BREG to EREG, provided, of course, that no Op Code or Interrupt has been encountered, and then resume the execution of data stored in EREG as before. The D_{xy} pulse also sends a DEMAND for the next data, which gets loaded in BREG. This is achieved by means of the SYNC latch, which is set by any OP code or Interrupt (except HALT, as explained later). SYNC latch is used to clear the BREG whenever it gets set as explained in Section 5.3.

(2) Execution of OP Codes:

OP codes are incorporated in the input data in a manner discussed in Section 5.4, in order to perform the PASS OVER,CYCLE OVER and JOB OVER operations, the corresponding codes being stored in the program at appripriate

places. We assume here a 2-bit code resulting in four different operating instructions as explained in Table 5.1.

Table 5.1: OP CODES

OP CODE	INTERPRETATION	ACTION TAKEN	CONDITION AFTER ACTION
01	PASS OVER	Retrace the path towards the DEFINED ORIGIN first along the y-axis and then along the x-axis. Increment the Pass counter.	The process continues
10	PASS OVER	Retrace the path towards the DEFINED ORIGIN first along the x-axis and then along the y-axis. Increment the PASS counter.	The process continues.
11	CYCLE OVER	Stop after the data in EREG is executed. Increment the CYCLE counter.	System can be restarted by manual START pulse.
00	JOB OVER	Stop after the data in EREG is executed. Give indication of JOB OVER. Increment CYCLE counter.	System has to be first RESET* and then only it can be restarted by manual START pulse.

* This feature is introduced to prevent the operator from accidentally starting the next job before setting up the machine and work-piece for the next job.

(3) Interrupt Facilities:

The interrupt commands HALT, STOP, INPUT ERROR (IE) and OUTPUT ERROR (OE) can over-ride the normal operation and are explained in the Table 5.2.

Table 5.2: INTERRUPT FACILITIES

CONDITION	ACTION	PROCEDURE FOR RESTART
HALT	Stop immediately with EREG and BREG holding the present data.	Manual START pulse
STOP	Stop after executing data in EREG and BREG	
IE	Stop after executing data in EREG.	Manual RESET* followed by Manual START pulse
OE	Stop immediately.	

* This feature is introduced to ensure that the operator corrects the error condition and resets the system before continuing with the job. However a built-in provision prohibits the system from running in case the operator presses the RESET without correcting the errors.

The manual RESET feature is necessitated in connection with JOB OVER, OP code as well as IE and OE interrupt conditions to permit the system to start by enabling READY latch to be set, because of the fact that these interrupt conditions inhibit the setting of READY latch, which is used to generate a single pulse whenever the manual start is present.

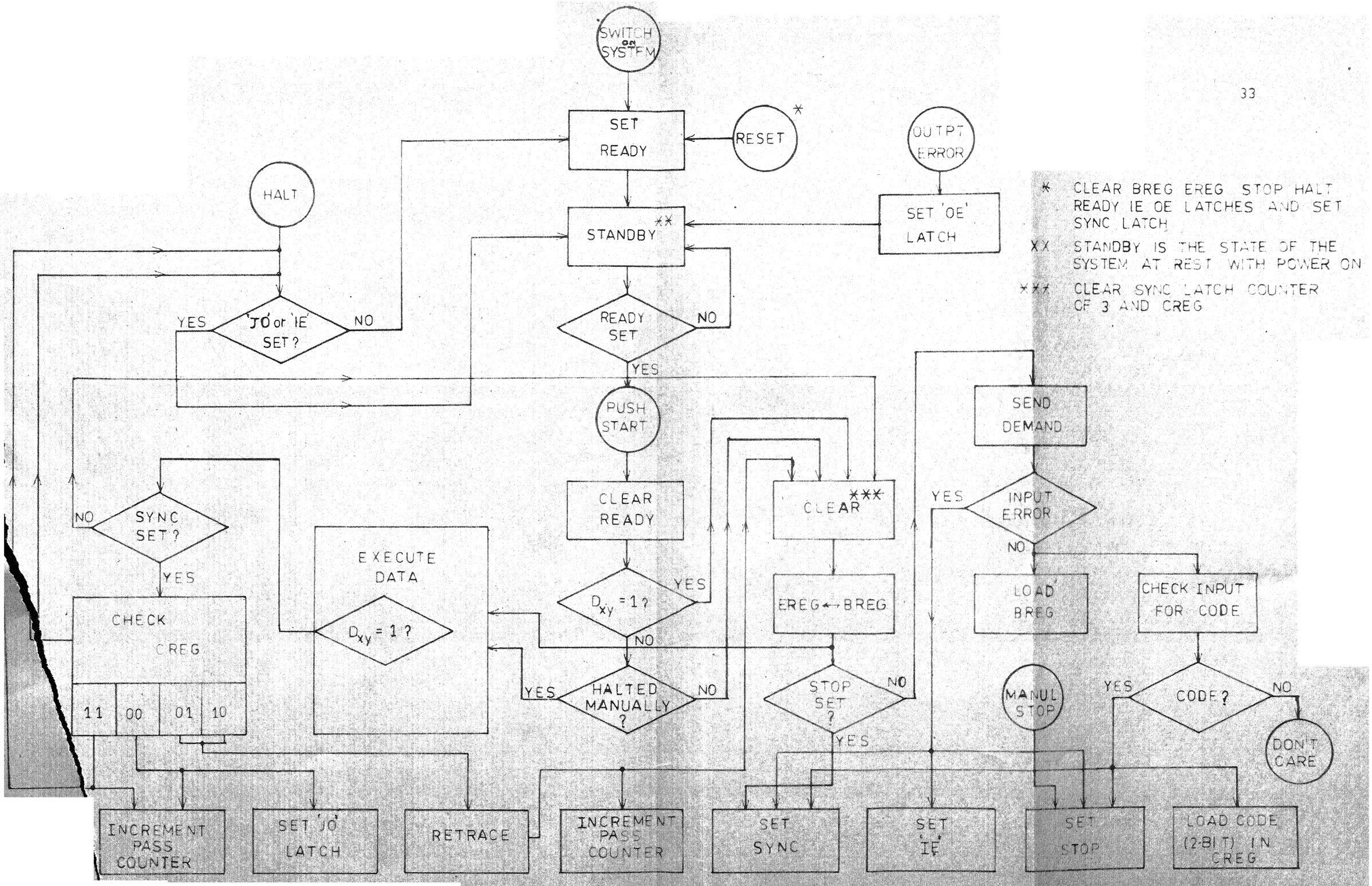
The complete flow-chart based on the foregoing considerations is shown in Figure 5.1. The STAND-BY block represents the state of the system whenever it is awaiting a manual intervention for re-start. The sequences under various conditions are now explained. These can be easily looked into the flow-chart.

A. INITIAL START

1. Switch ON system.
2. READY latch is set.
3. Push START.
4. READY latch is cleared.
5. SYNC latch, counter of 3 and CREG are cleared.
6. Data is loaded from BREG to EREG.
7. DEMAND is sent.
8. BREG is loaded with data.
9. Steps 6,7 and 8 are repeated.
10. Data stored in EREG is executed.

B. CONTINUOUS RUN

1. Push START.
2. READY latch is cleared.
3. SYNC latch, counter-of-3 and CREG are cleared.
4. Data is loaded from BREG to EREG
5. DEMAND is sent.
6. Data is executed until $D_{xy} = 1$
7. Sequence 3-4-5-6 is repeated.



* CLEAR BREG EREG STOP HALT READY IE OE LATCHES AND SET SYNC LATCH

XX STANDBY IS THE STATE OF THE SYSTEM AT REST WITH POWER ON

*** CLEAR SYNC LATCH COUNTER OF 3 AND CREG

FIG-5.1 FLOW CHART

C. PASS OVER

1-6 Same as in CONTINUOUS RUN

7. Code is sensed, hence STOP and SYNC latches are set.

8. CREG is checked.

9. The table is brought to the DEFINED ORIGIN.

10. The PASS counter is incremented.

11. System goes to step 3.

D. CYCLE OVER

1-8 Same as in PASS OVER.

9. CYCLE counter is incremented.

10. READY is set.

11. Go to step 1.

E. JOB OVER

1-8 Same as in PASS OVER.

9. CYCLE Counter is incremented and JOB OVER indication is lighted.

10. READY latch is not set and system comes to STAND-BY.

F. INPUT ERROR

1-6 Same as in CONTINUOUS RUN.

7. INPUT ERROR is sensed, STOP latch and SYNC latches are set and INPUT ERROR indication is lighted.

8. CREG is checked.

9. No code. Ready latch is not set and system comes to STAND-BY.

G. OUTPUT ERROR

1-6 Same as in CONTINUOUS RUN.

7. OUTPUT ERROR sensed.

8. OUTPUT ERROR indication is lighted.

9. READY is not set and system comes to STAND-BY.

H. HALT

1-6 Same as in CONTINUOUS RUN.

7. Push HALT

8. HALT indication is lighted.

9. READY latch is set and system comes to STAND-BY.

J. STOP

1-6 Same as in CONTINUOUS RUN.

7. Push STOP.

8. STOP indication is lighted.

9. SYNC latch, counter-of-3 and CREG are cleared.

10. Data is loaded from BREG to EREG.

11. SYNC latch is set and hence BREG is cleared.

12. Data in EREG is executed until $D_{xy} = 1$.

13. CREG is checked.

14. No code, hence READY latch is set and system comes to STAND-BY.

5.2 DEMAND PULSE GENERATOR

It has already been pointed out that the Serial Access Memory supplies data word by word. As one demand pulse is

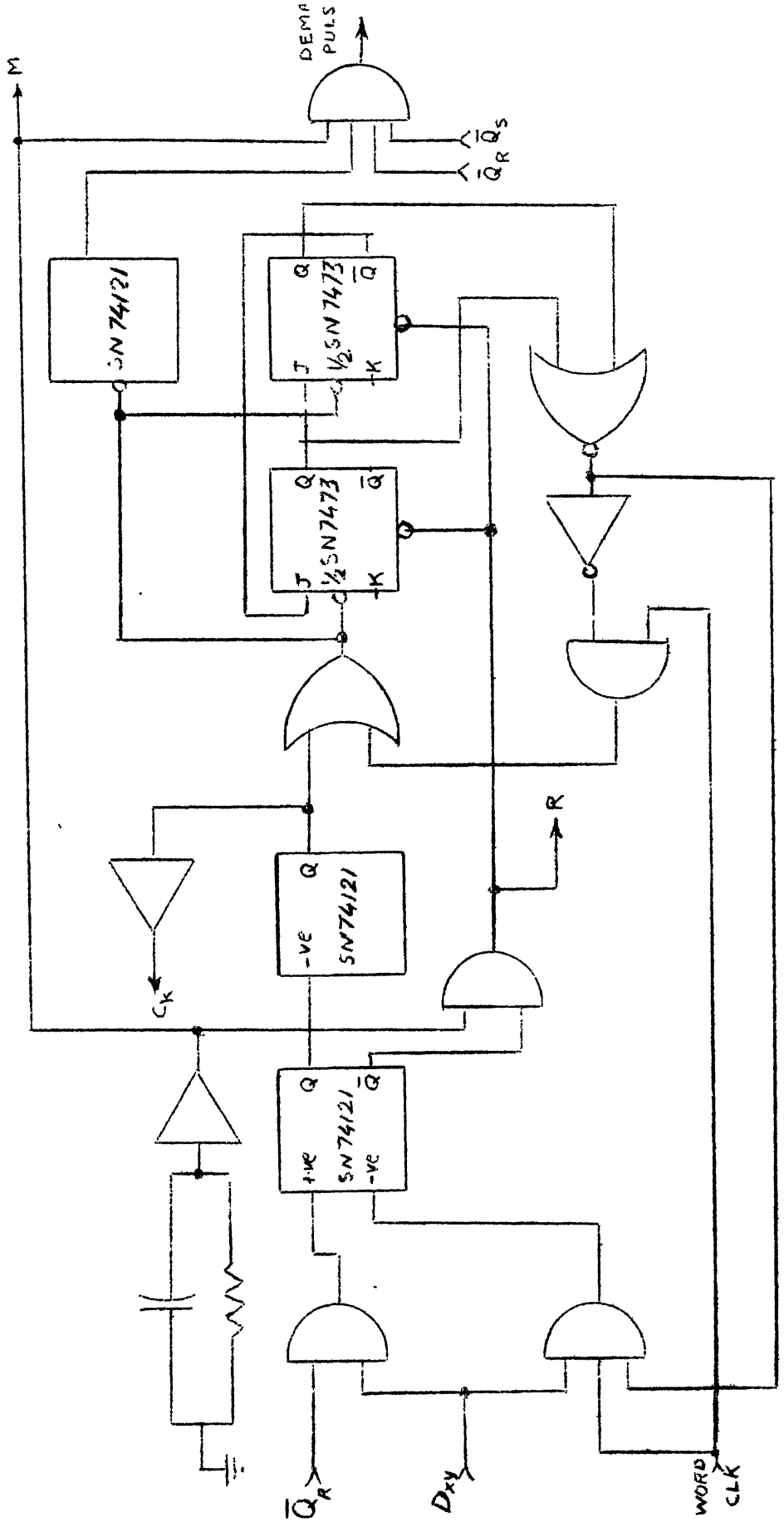


FIG-5.2 DEMAND PULSE GENERATOR

is required for each word hence three demand pulses are required for each interpolation interval which necessitates a counter of three for the purpose. When the system control is switched on, both the EREG and the BREG are cleared automatically. With initial starting of the process, the BREG is filled with data and a set of three demand pulses is automatically generated to obtain the next data also. For performing other operations - clearing SYNC latch, CREG etc. as explained in the FLOW-CHART, before the next set of demand pulses is sent to the SAM, two monostable multis have been used. The circuit realisation for the demand pulse generator is shown in Figure 5.2.

5.3 INTERRUPT CIRCUITRY AND SYNC LATCH

As has been pointed out in Section 5.1, the system can be interrupted by either of the following commands or conditions:

(i) HALT command given through a manual push-button,

(ii) STOP command, given through a manual push-button,

(iii) INPUT ERROR conditions, and

(iv) OUTPUT ERROR condition.

The HALT, STOP, INPUT ERROR and OUTPUT ERROR latches store the respective bits of information regarding the interrupt mode called for. It is essential to clear all these latches whenever the process is to be started. The START switch has been interlocked and is effective only when the READY latch is set (Figure 5.3). Before a demand pulse is sent to the

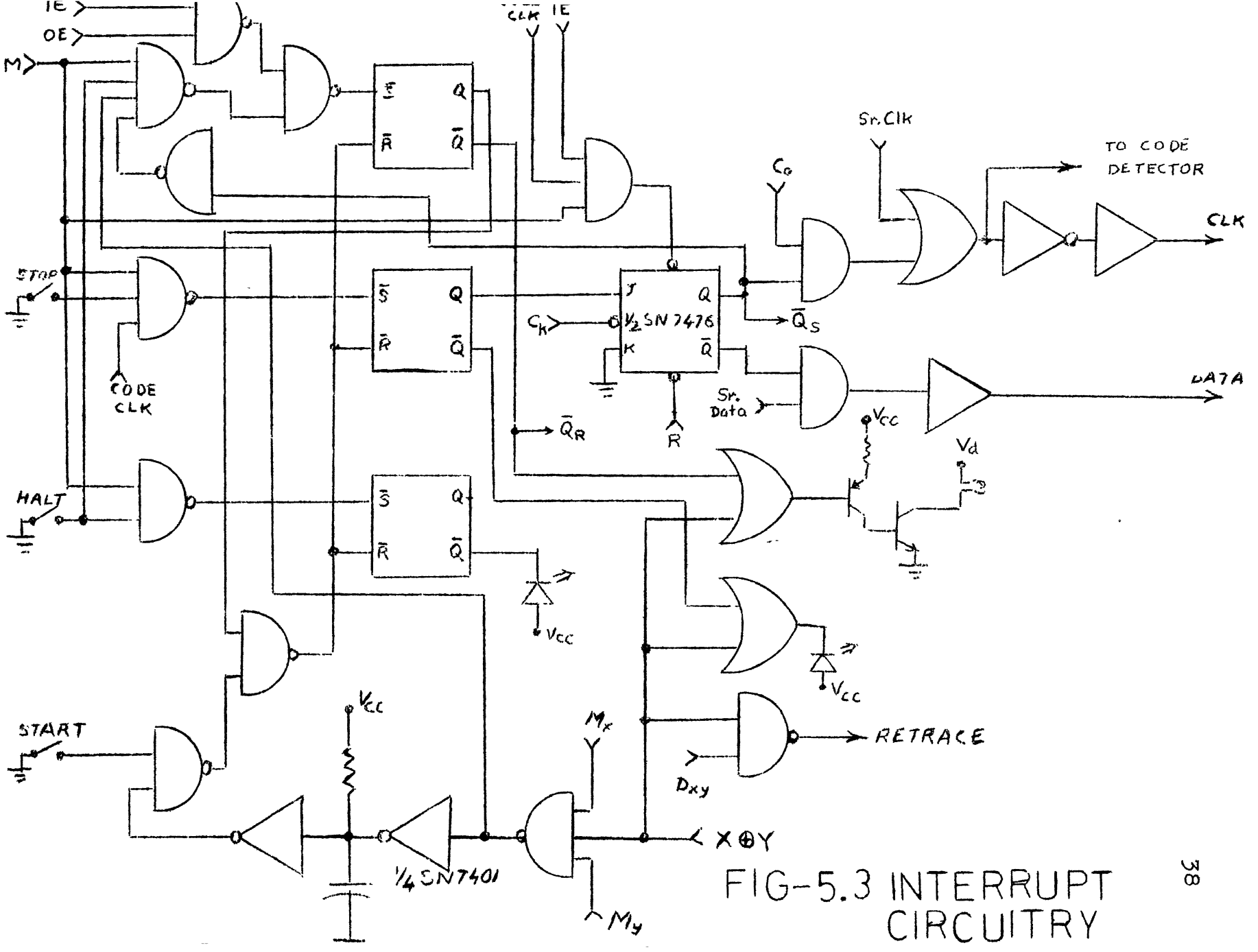


FIG-5.3 INTERRUPT CIRCUITRY

SAM, different operations have to be performed in a sequence given below:

(i) The SYNC latch and the CREG are to be cleared.

(ii) The data is to be transferred from the BREG to EREG.

(iii) If the STOP latch is set, then the SYNC latch is to be set which inhibits the demand pulse and clears the BREG.

To achieve the above sequence, as pointed in Section 5.1, two monostable multis are used. Automatic clearing of the system and setting appropriate conditions of the latches have been achieved by a time delay circuit whenever the CONTROL is switched ON. The SYNC latch is used to clear the BREG whenever it is set.

5.4 CODE DETECTOR AND CODE REGISTER

The code is extracted from the data which is supplied to the system as explained here. The data for every interpolation interval consists of three words, each of twelve bits, the last bit in every word being the parity bit. Also, the length of the cut for any interpolation interval can never be zero. Hence if the first eleven bits in any interpolation data of 36 bits are zero, the data is considered to be a code. This is achieved by using the word-clock. In Figure 5.4 two cases of data have been considered: (a) When at least one data bit in the first word is '1'. (b) When none of the data bits is '1'. The

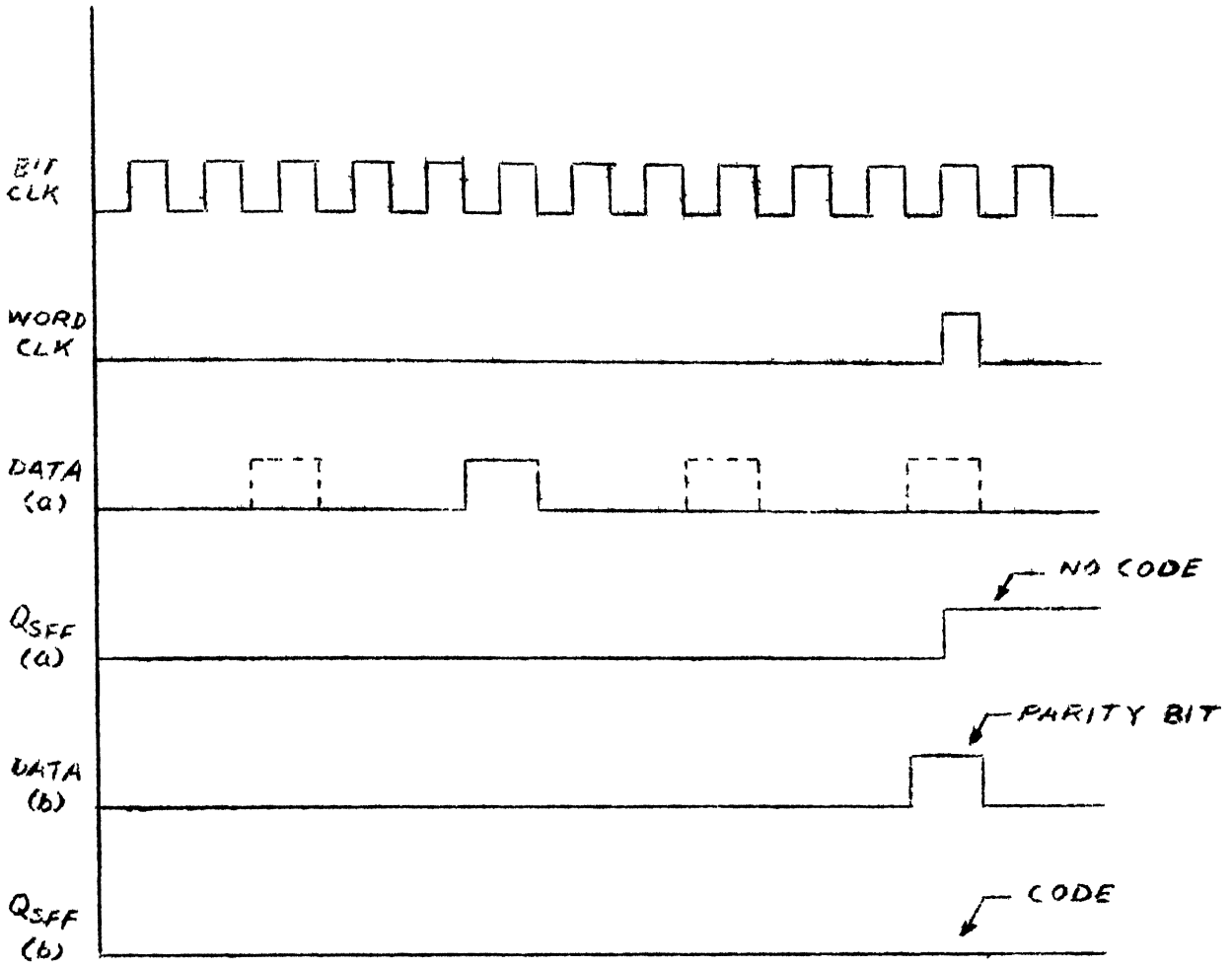


FIG-5.4 TIMING DIAGRAM

parity bit has not to be considered, avoided by word clock which senses and stores the information in a flip-flop (Q_{SFF}) before the arrival of parity bit. The case (a) depicts that next two words are not coded data and (b) indicates that the next two words are coded. The two-bit code is again extracted by the same principle as the code sensing. The rest of the 24 bits (i.e. two words of data) are similarly checked. If in either word all the data bits except the parity bit are zero then the code register takes the code bit as '0' else it takes the code bit as '1'. It should be noted that the CLOCK to the CREG is allowed only when the code indication has been marked. When the data in EREG is executed, the next operation is commanded by the code stored in the CREG. The circuit realisation is shown in Figure 5.5.

5.5 GATING CIRCUIT

This gating circuit is required for providing the facility of 'fast return' to the 'defined origin', i.e. to retrace the path, when the RETRACE code commands it to do so. A decoder is, in general, needed for each axis to detect that the machine table has reached 'zero' position for the respective axis. The SN74190 up/down decade counters are used to track the position of table for each axis which give a decoded output (Max./Min.) going high when it has contents '0' and is in down counting

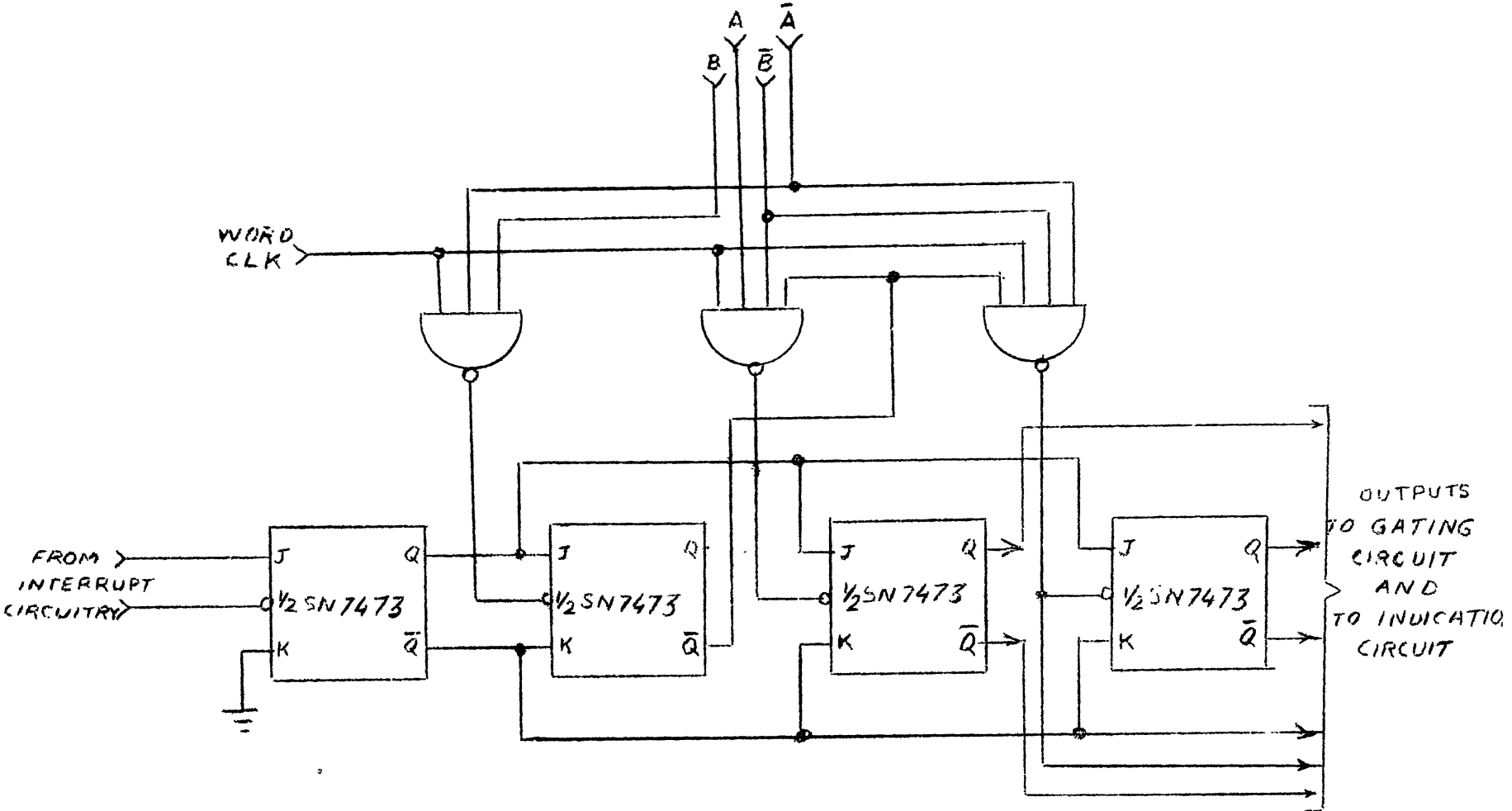


FIG-5.4 CODE DETECTION AND STORAGE CIRCUIT

mode or has contents '9' and is in up counting mode. As in present set-up the counters have to be in the down counting mode to be able to reach the origin, these decoded outputs of all counter stages are AND-ed to give decoder output corresponding to that axis thus eliminating a separate decoder. These have been called M_x and M_y. The gating circuit uses these outputs to provide the proper controls (CLOCK SELECT and GATE SELECT, which are explained in the next chapter) to the stepper motor translator so that the Defined Origin is reached. Depending on the RETRACE code, the gating circuit permits the movement in the appropriate axis first and then in the other axis. The circuit realisation is shown in Figure 5.6.

5.6 INDICATION CIRCUIT

This circuit is realised for the detection of INPUT ERROR and, it's as well as job over indication. The input error is sensed if the data does not come from the SAM in spite of the demand within a specified time (built-in the hardware depending on the design). This error sets a latch and gives visual indication on the console, the other role played by this error has already been explained in Section 5.1. With the completion of the job, the JOB OVER signal on the console is lighted and the READY latch is inhibited from being set. This signal also increments the CYCLE OVER counter. The CYCLE OVER Counter is incremented also when

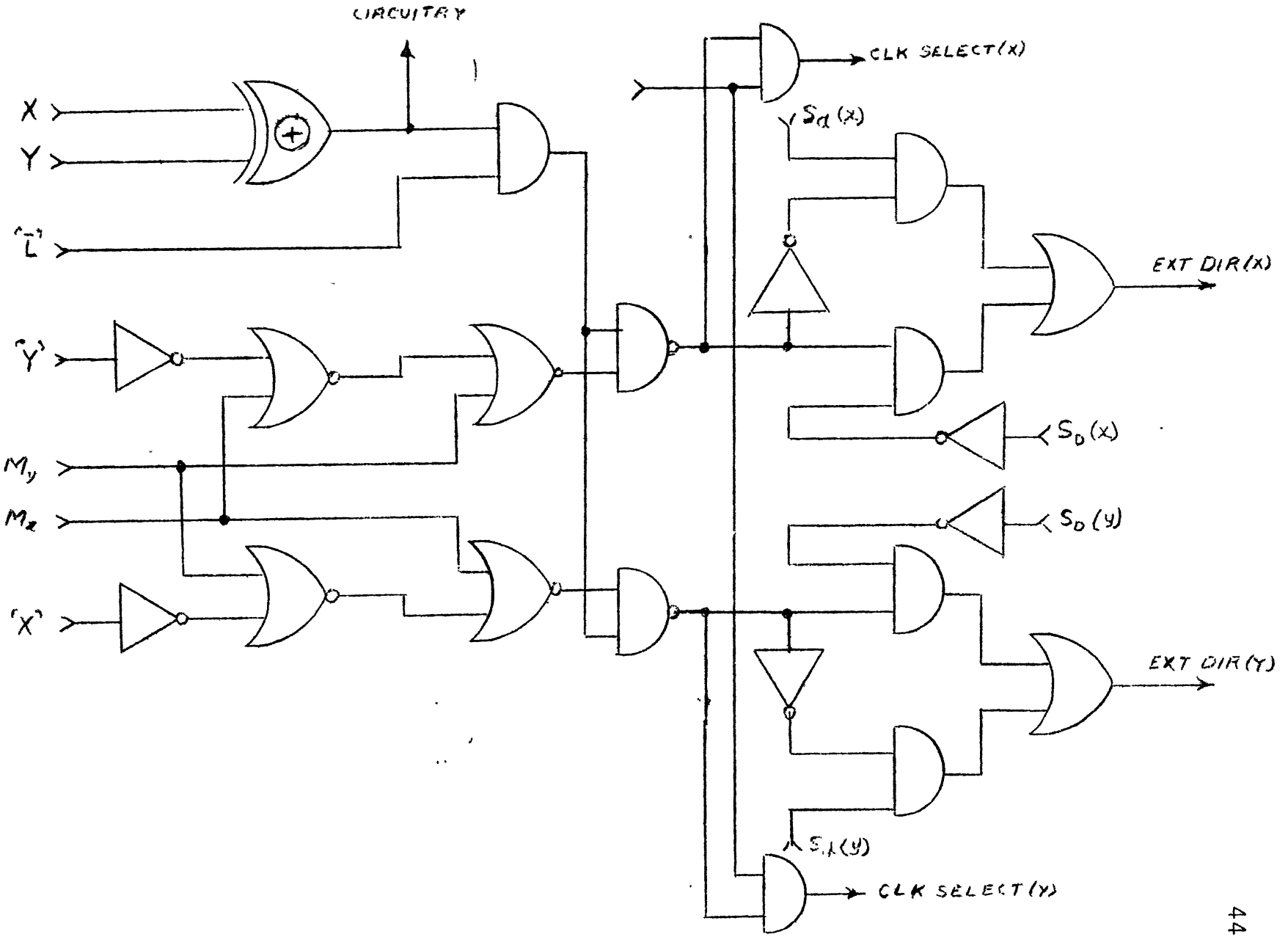


FIG-5.6 GATING CIRCUITRY

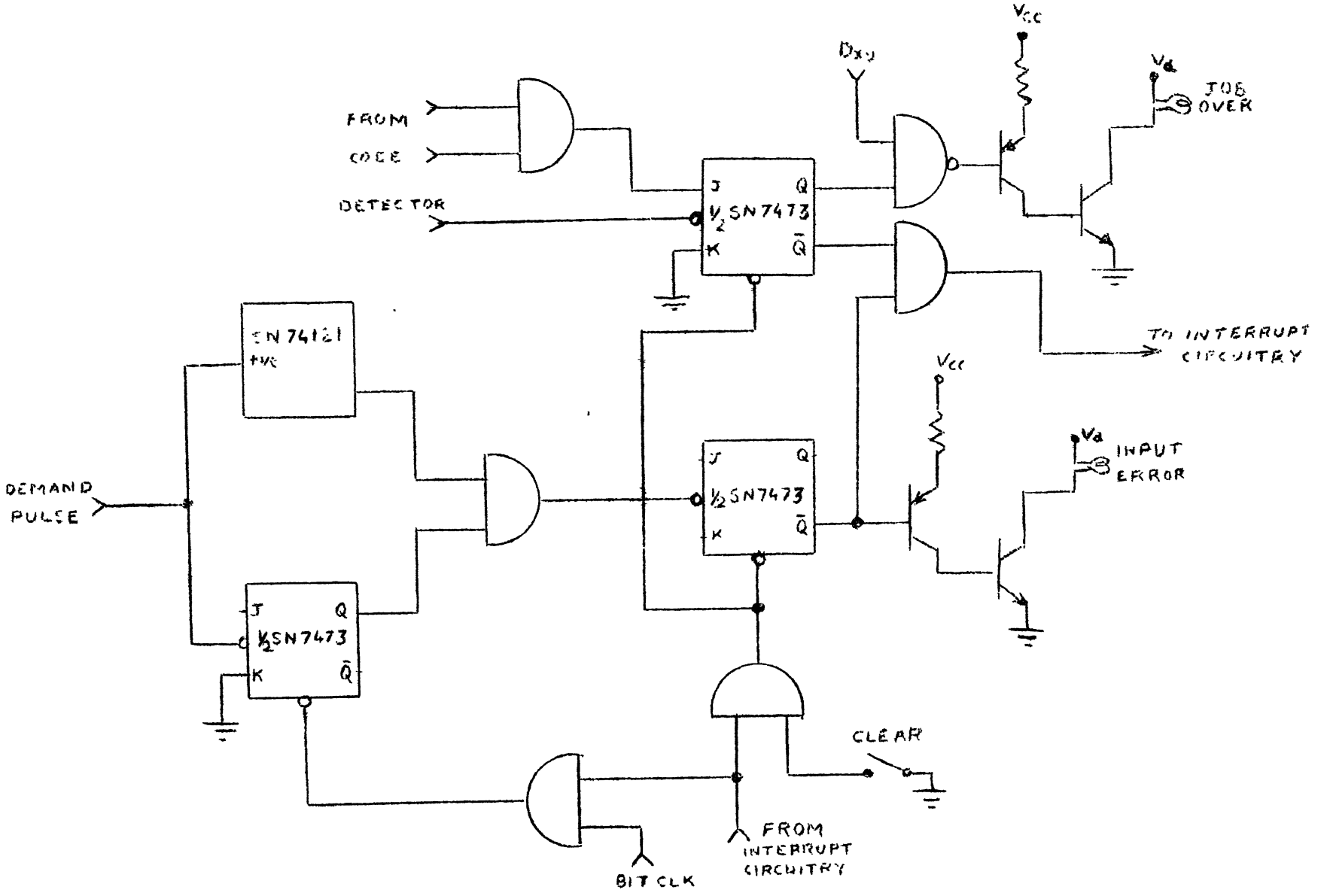


FIG-5.7 INDICATION CIRCUITRY

its code is sensed and the execution of the data in EREG is completed. The circuit realization is shown in Figure 5.7. These latches can be cleared and the system enabled by the push-button RESET.

5.7 FEED CLOCK GENERATOR

A very simple circuit is used for the feed-clock generator used to provide a variable clocking range to vary the feed-velocity of the table. The circuit uses a proper diode simulated by two transistors, as shown in Figure 5.8. The waveform generated by the negative resistance relaxation oscillator is shaped by a Schmitt trigger.

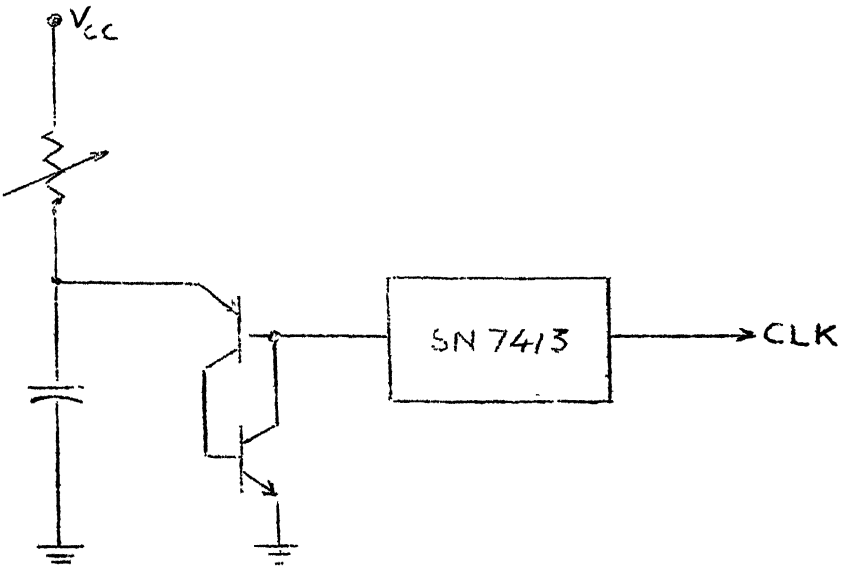


FIG-5.8 FEED CLOCK GENERATOR

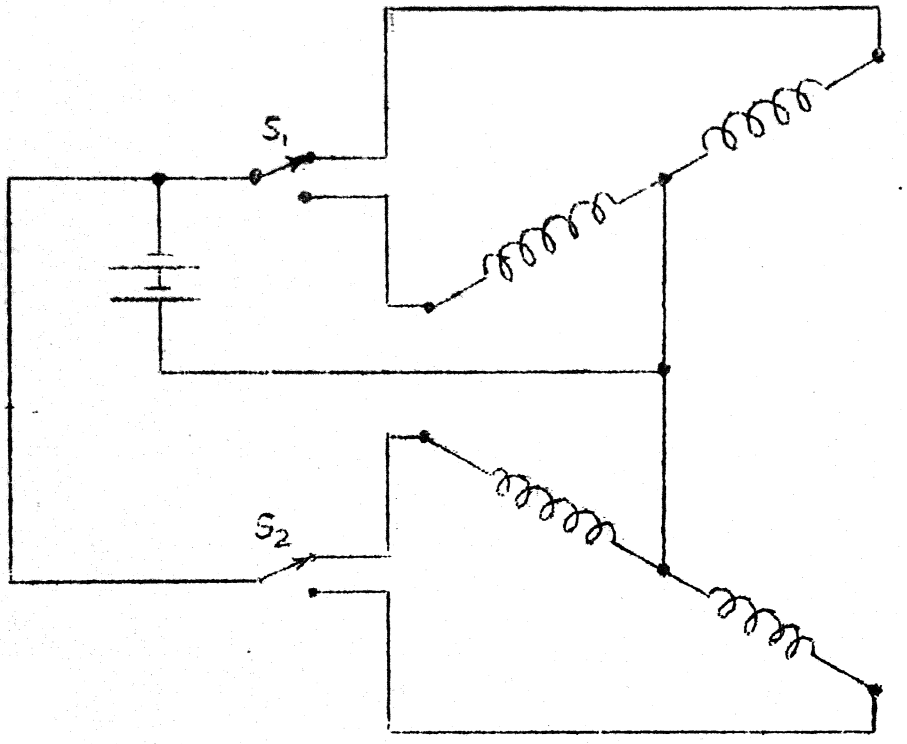
CHAPTER 6

STEPPER MOTOR TRANSLATOR

This chapter deals with the translator which is used to drive the stepper motor. After discussing the specifications of the stepper motor used in the system, the generation of the sequence required by the stepper motor is taken up. The third section presents the control-logic requirements of the translator. Then the single-pulse and clock-burst generator is taken up and the chapter is concluded with a description of the power stage.

6.1 STEPPER MOTOR

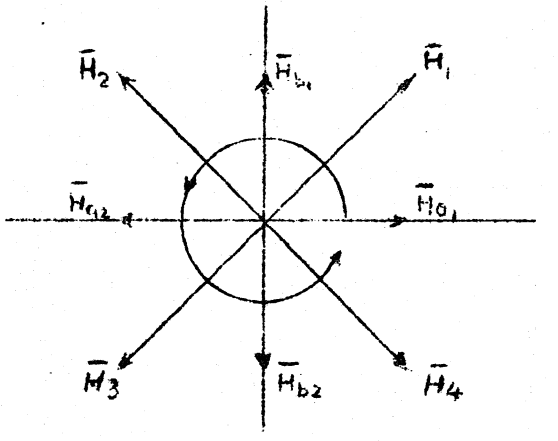
There are three broad classes of stepper motors (i) Permanent Magnet (ii) Variable Reluctance and (iii) Hybrids. 'Permanent Magnet' type is the most common. The one that has been used in the present work is of the permanent magnet type with bifilar windings. In this type the stator winding corresponding to each pole-pair consists of two centre-tapped coils wound in opposite direction (Figure 6.1(a)). At any time, only one half of either winding carries current and the motor movement takes place by alternate switching of current from one half of the winding to the other half of the same winding. The four states depending on the positions of the two switches S_1, S_2 are given in Figure 6.1(b). Figure 6.1(c) shows the



(a) CONFIGURATION

(b) SWITCHING SEQUENCE

STATE	$S_1 S_2$	ENERGISED WINDING SECTIONS
1	00	$a_1 b_1$
2	01	$a_1 b_2$
3	11	$a_2 b_2$
4	10	$a_2 b_1$



(c) VECTOR DIAGRAM

FIG-6.1 PRINCIPLE OF STEPPER MOTOR

directions of the magnetic field in these four states, indicated by $\overline{H}_1$, $\overline{H}_2$, $\overline{H}_3$ and $\overline{H}_4$ respectively, $\overline{H}_{a1}$, $\overline{H}_{a2}$ and $\overline{H}_{b1}$, $\overline{H}_{b2}$ being the magnitudes due to the individual coils. The direction of the movement depends on the sequence of these states, e.g. the sequence 1 2 3 4 results in the clockwise rotation of the magnetic field in the plane of Figure 6.1(c) while sequence 4 3 2 1 leads to the counter-clockwise rotation. The specifications of the motor are given below:

DC voltage = 12 volts

Current per winding = 1.25 Amps.

Torque = 3 Kg-cm

Pulse repetition rate should be 200 C/s

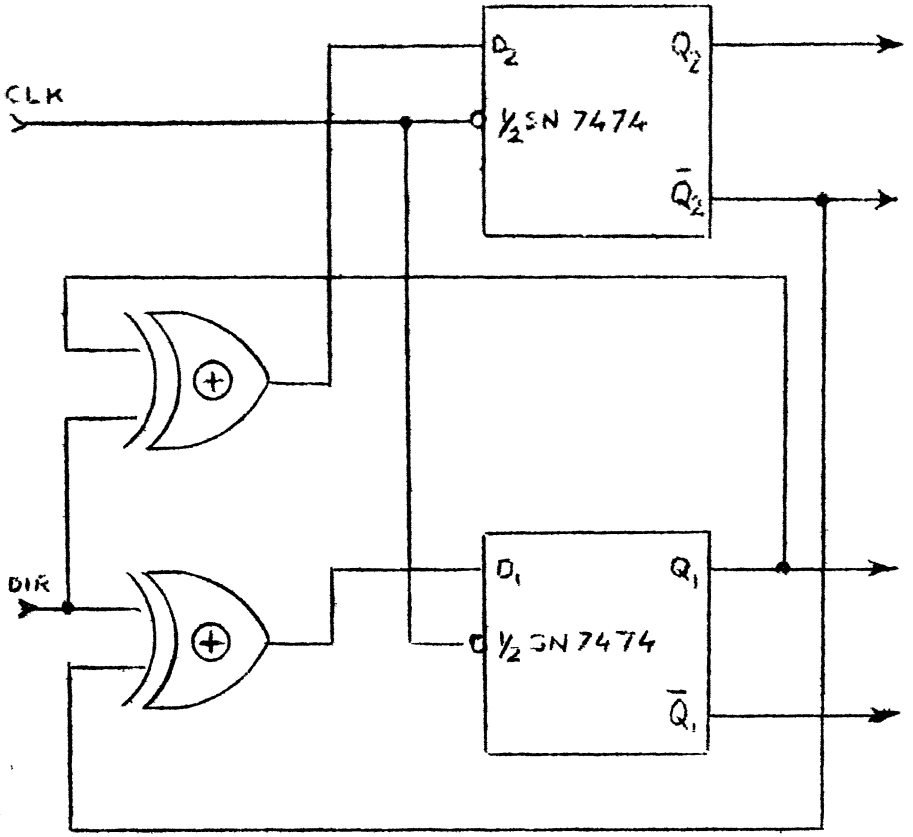
Pulse height (H) should be $5V < H < 12$ V

6.2 SEQUENCE GENERATOR

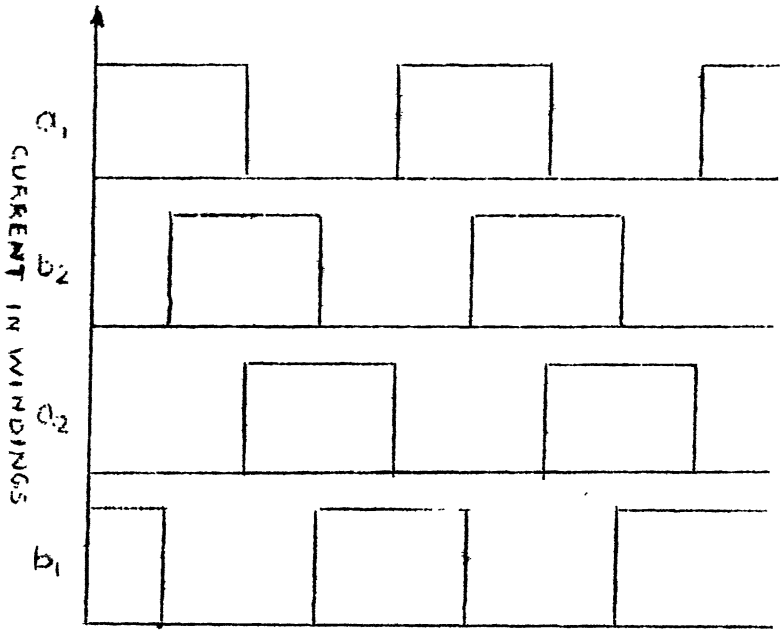
The switching sequence required for a stepper motor with bifilar winding has been described in the previous section. In actual practice, the information fed to move the stepper motor is the clocking (stepping) frequency and the direction in which it has to be moved. The sequence required for the two windings of the stepper motor are derived from this information and the circuit realization for the same is shown in Figure 6.2(a) with the phase currents in the half-windings shown in Figure 6.2(b).

6.3 CONTROL LOGIC

The requirement of the control logic for the stepper motor as required for the present work, are the manual



(a) CIRCUIT



(b) PHASE CURRENTS

FIG-6.2 SEQUENCE GENERATOR

operation and the automatic operation. In the manual operation, the direction and gating of the clock for the stepper motor movement has to be done by switches on the console, while for the automatic operation, an externally gated clock has to move the motor in a direction determined by an external direction information. This choice of the mode of operation is achieved by means of the control variable called MODE. With a view to making a general purpose translator for driving stepper motors, additional facilities have been incorporated. A control variable named CLOCK SELECT permits the user to select either an internal clock or an external clock, and another control variable CLOCK GATE allows the selected clock to be gated.

In manual operation, clock gating is achieved by a push-button called SLEW. It permits the clock as long as it is kept pressed. As manual gating cannot be perfect, another control (STEP) is provided which sends one and only one pulse whenever it is pressed. These two manual controls, are similar to CLOCK GATE and CLOCK SELECT in the remote mode. Directional information is given by a switch for manual mode and by a one-bit information in the remote mode. The realisation is shown in Figure 6.3.

6.4 SINGLE-PULSE AND CLOCK-BURST GENERATOR

The internal clock generator for a translator has to fulfil two requirements, viz. (1) Provide a continuous clock, (2) Provide a single pulse in response to a manual

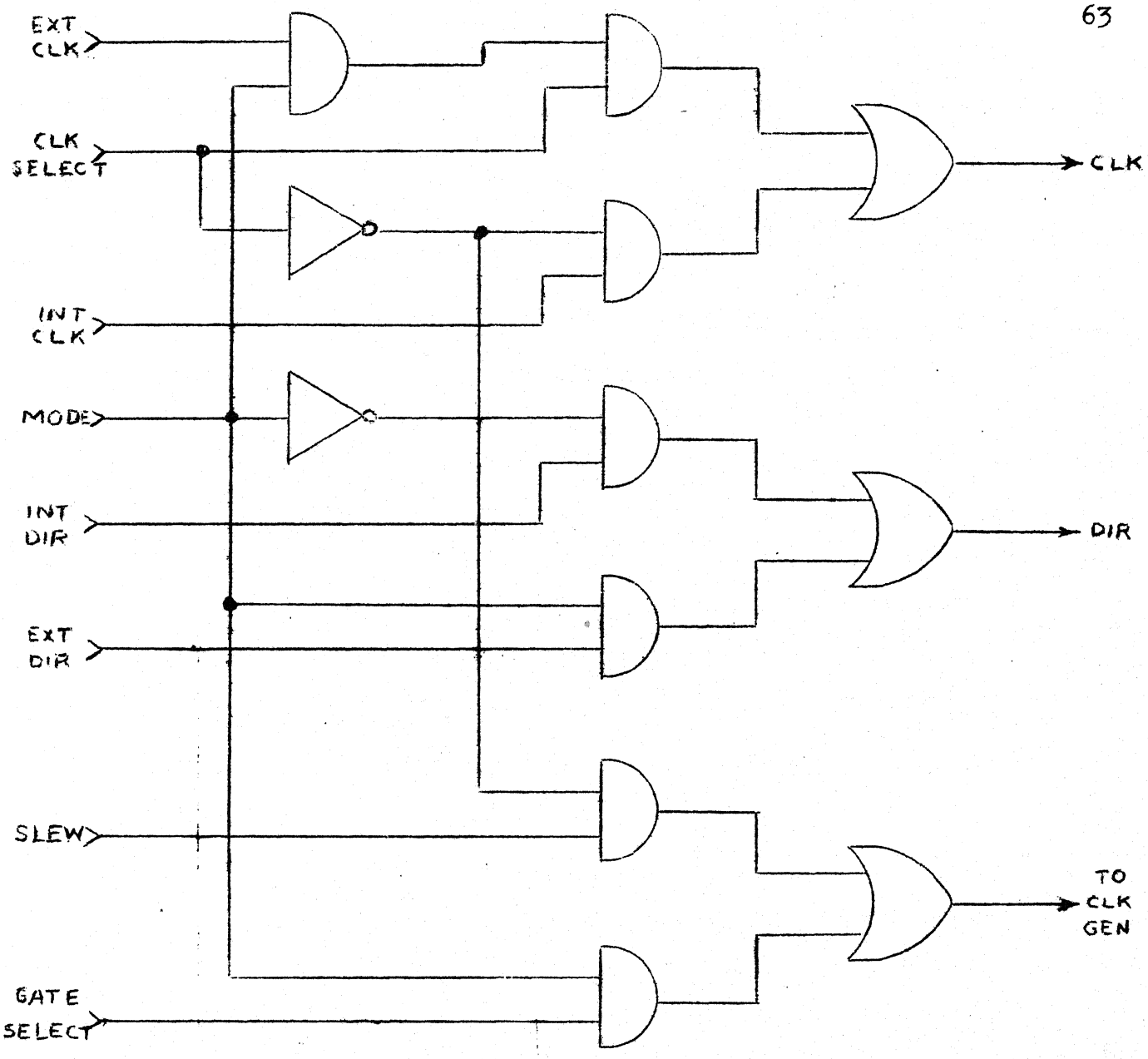


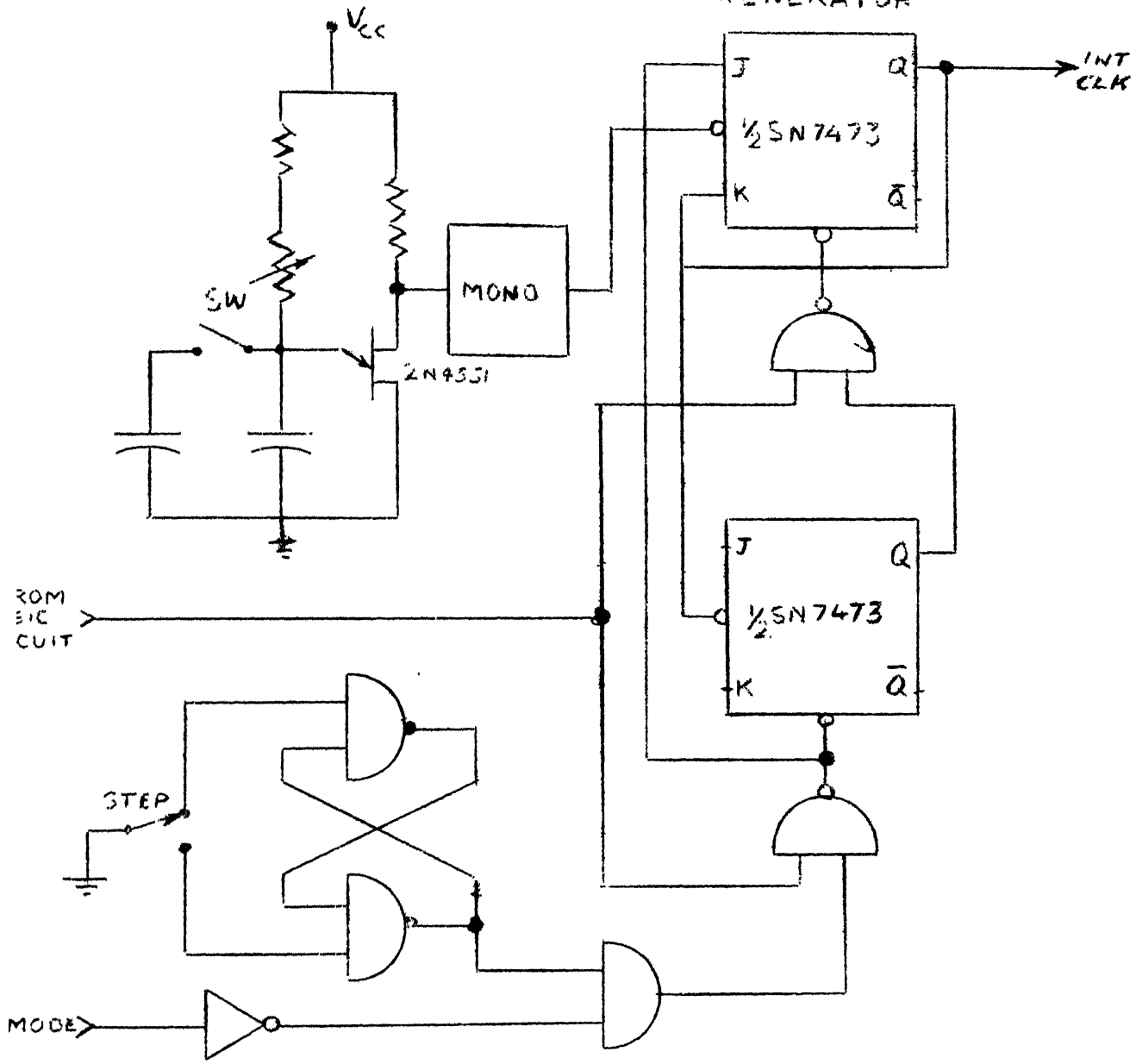
FIG-6.3 LOGIC CIRCUIT

push-button. The clock frequency should be linearly variable from about 2 Hz to 200 Hz. This has been realized using a UJT oscillator, two J-K flip-flops and a few gates as shown in Figure 6.4. The complete clock range is obtained by putting another capacitor in parallel by a switch SW. Linear range for each position of the switch is obtained by varying the resistance in the R-C timing circuit.

6.5 POWER STAGE

Each half of the two windings of the motor has to be driven by a stage, which is simply a power-transistor switch preceded by a TTL - compatible pnp driver, as shown in Figure 6.5. The function of the 'free-wheeling' diode placed in parallel with the winding is to allow the current through the winding to decay gradually after the power stage is switched off.

FIG-6.4 SINGLE-PULSE AND CLOCK BURST GENERATOR



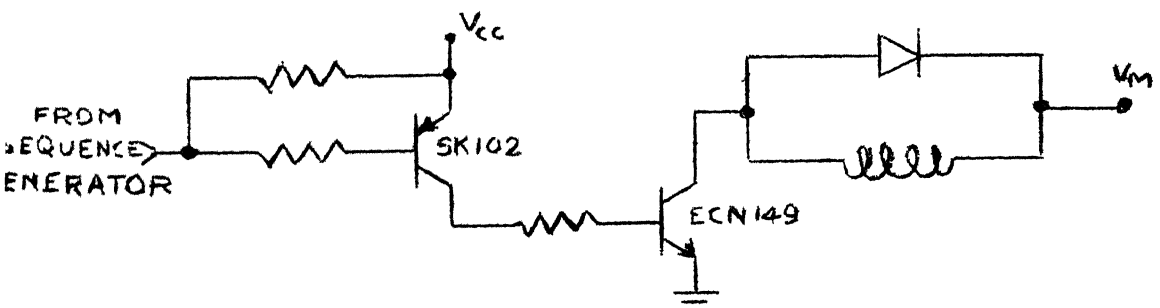


FIG-6.5 POWER STAGE

CHAPTER 7

LAYOUT AND OPERATING INSTRUCTIONS

This chapter describes the card-layouts, different controls and indications on the console. Then a brief description of its operation is given.

7.1 CIRCUIT LAYOUTS

The whole circuit is spread in eleven printed circuit cards classified in four heads. Table 7.1 gives functional description of cards and Table 7.2 gives the card layouts.

7.2 POWER SWITCHES

(i) MAINS - The main switch for switching ON 230V AC (1 Amp.).

(ii) MOTOR - The motor switch for enabling power to the stepper motors.

(iii) CONTROL - This switch energises the control system.

7.3 PANEL DISPLAYS

(i) 4 digit position display along x and y axes

(ii) Passes completed (2-digit)

(iii) Cycles completed (2-digits)

(iv) Retrace LED indicating movement under PASS code to reach the defined origin.

Table 7.1: FUNCTIONAL DESCRIPTION OF CARDS

CLASS	CARD NAME	DESCRIPTION	CARD NO.
SYSTEM	BUFFER-DIVIDER DECODER	PRM clock generator incorporating inhibition of clock if interpolation interval is zero. BREG and EREG for C.	7.1
	BUFFER-PRM-COUNTER	BREG, EREG and PRM's for x and Y.	7.2
	BUFFER-PRM-COUNTER	Counter for x and y.	
CONTROL	CONTROL-I	Demand clock generators, Sync. latch, code detector, CREG, IE Latch	7.3
	CONTROL-II	Gating circuitry for RETRACE, Feed clock generator, Interpolation Flip-flop.	7.4
	CONTROL-III	HALT, STOP, READY Latches and corresponding indications.	7.5
TRANSLATOR	X-Drive Y-Drive	Sequence generator, single pulse and burst generator Gating circuit stage.	7.6
DISPLAYS	X-DISPLAY Y-DISPLAY	4 digit position display, counters and decoder drivers, logic circuit for display.	7.7
	PASSES AND CYCLES DISPLAY	PASS counter, CYCLE Counter with corresponding decoder-drivers and 2-digit displays. JOB-OVER Indication.	7.8

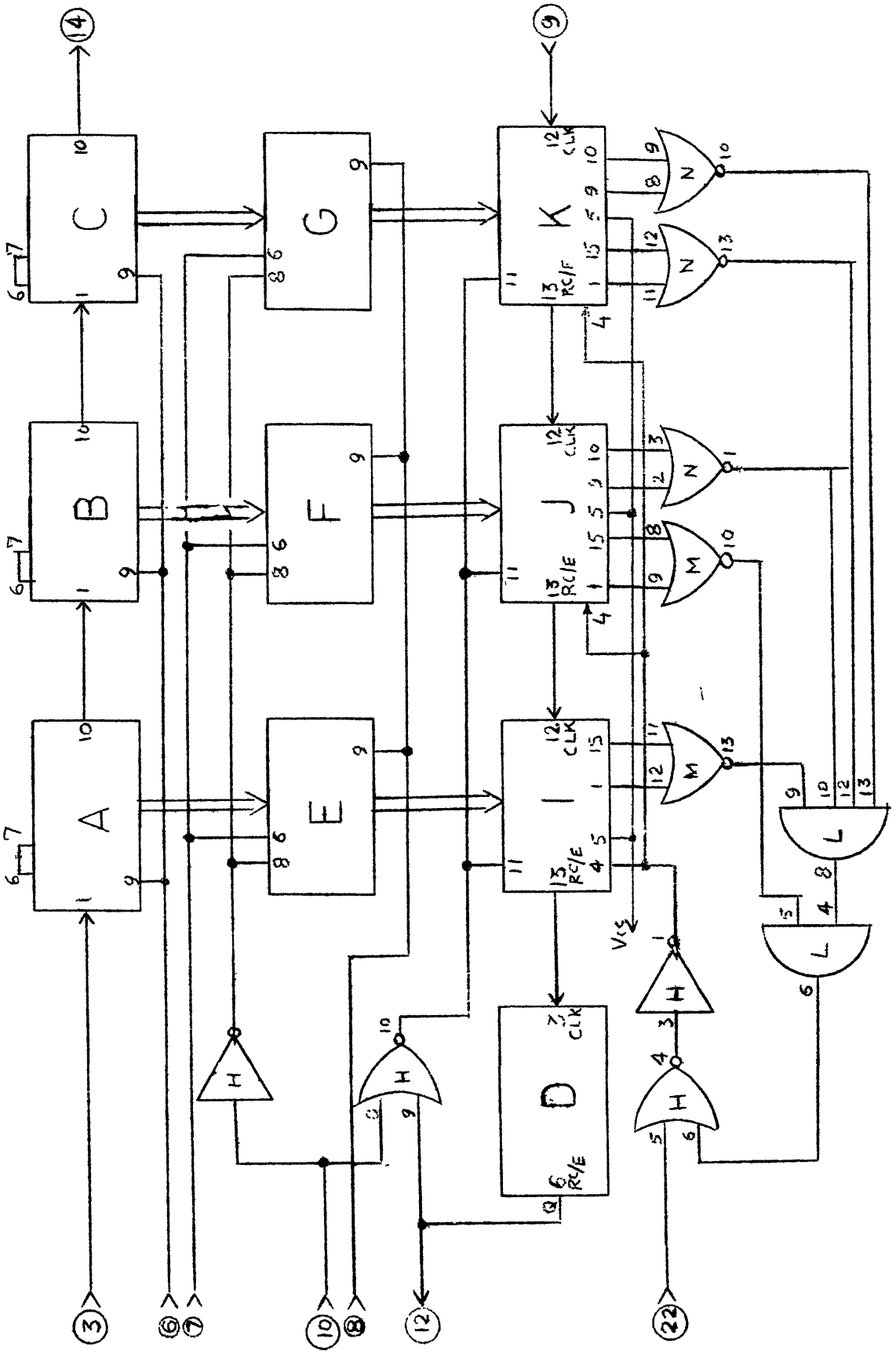


FIG-7.1 BUFFER DIVIDER DECODER

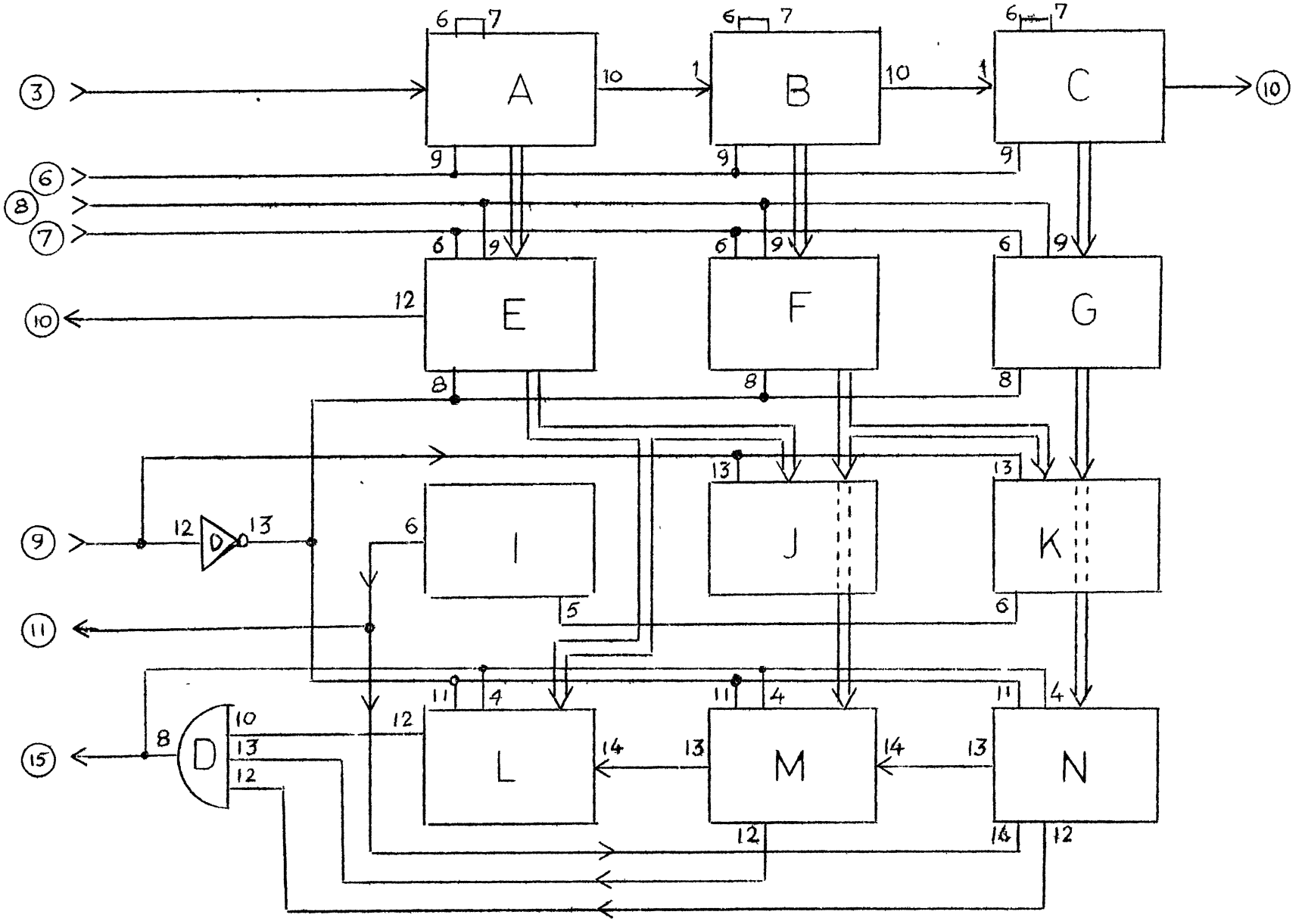


FIG-7.2 BUFFER PRM COUNTER

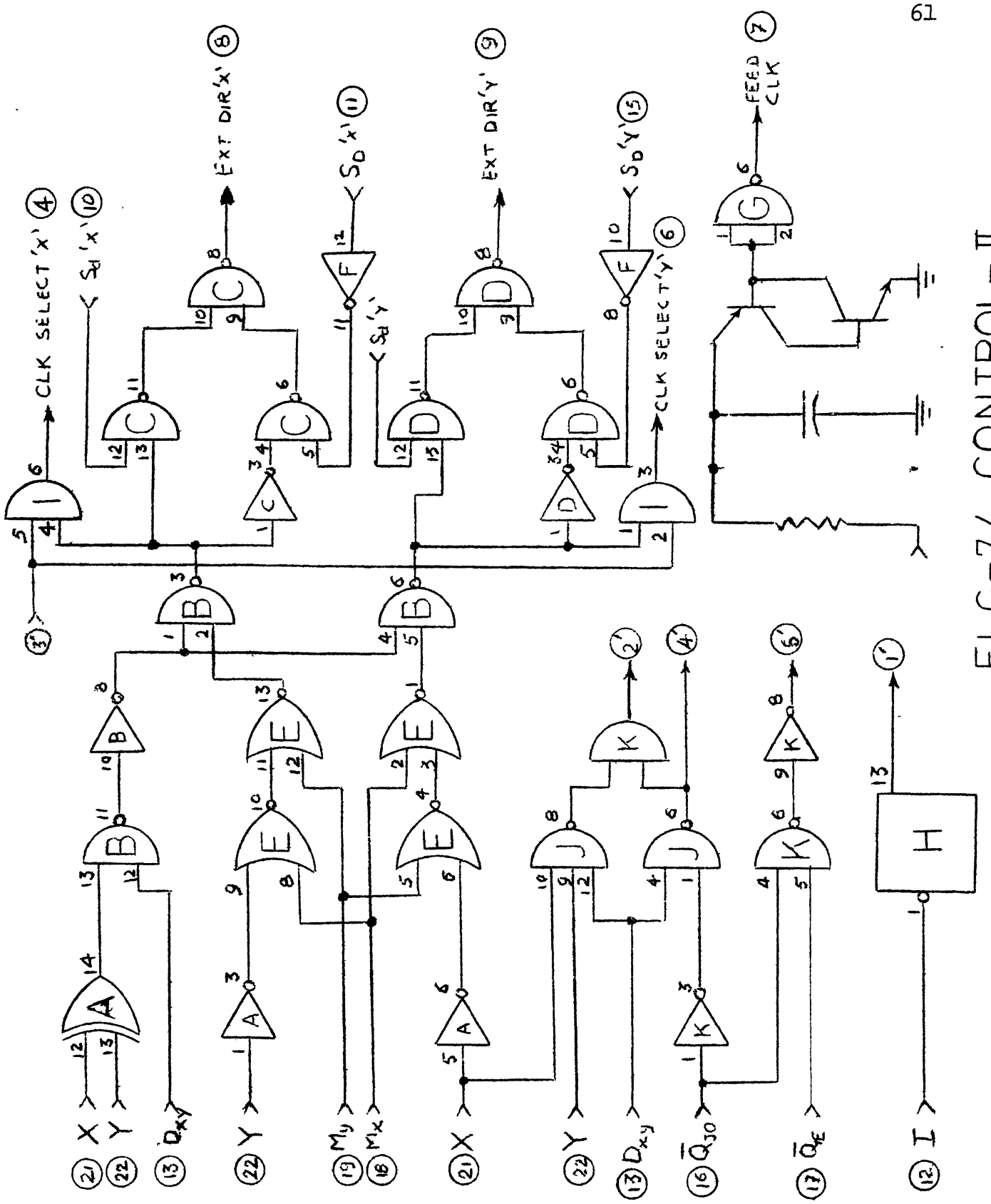


FIG-7.4 CONTROL-II

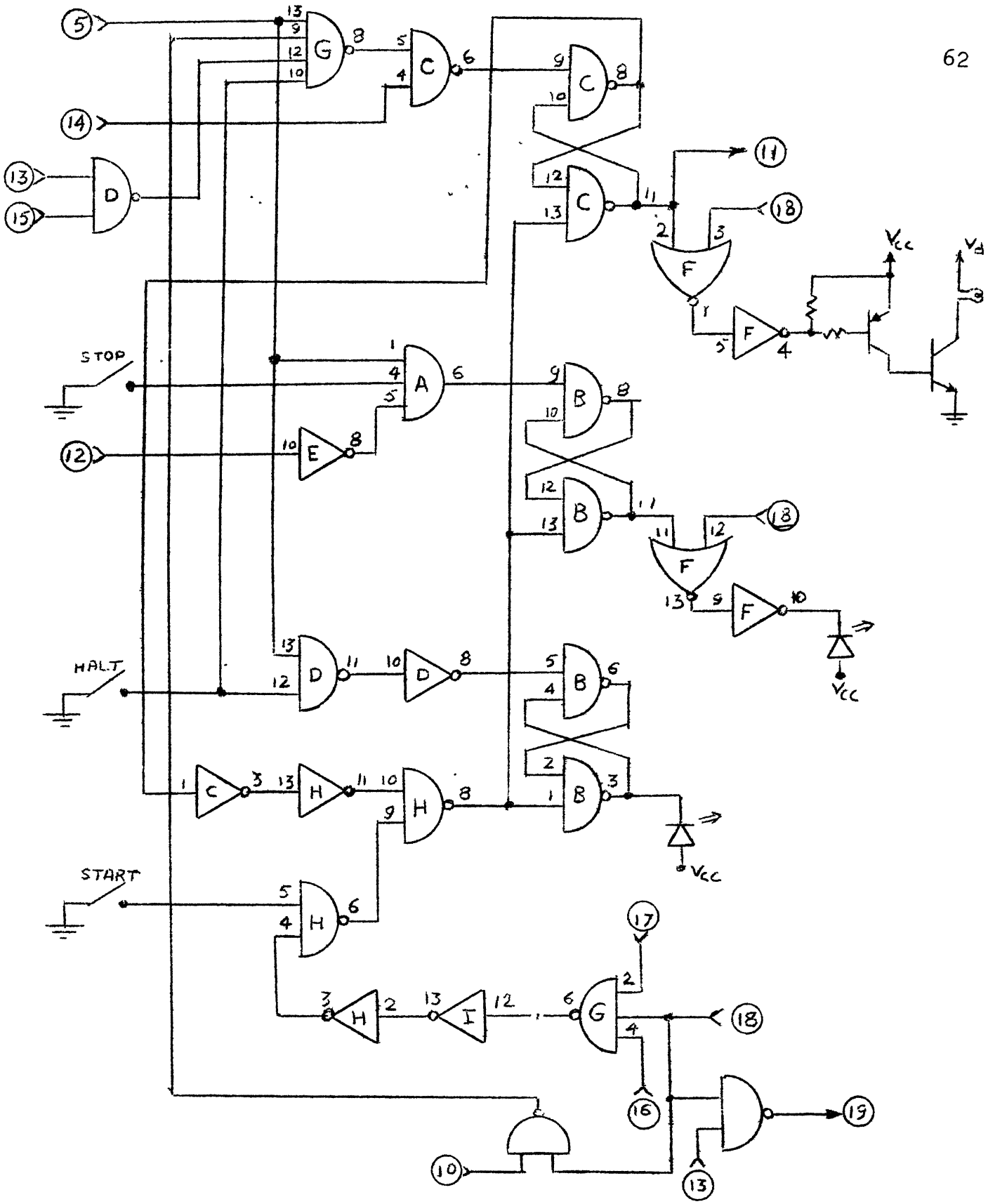


FIG-7.5 CONTROL - III

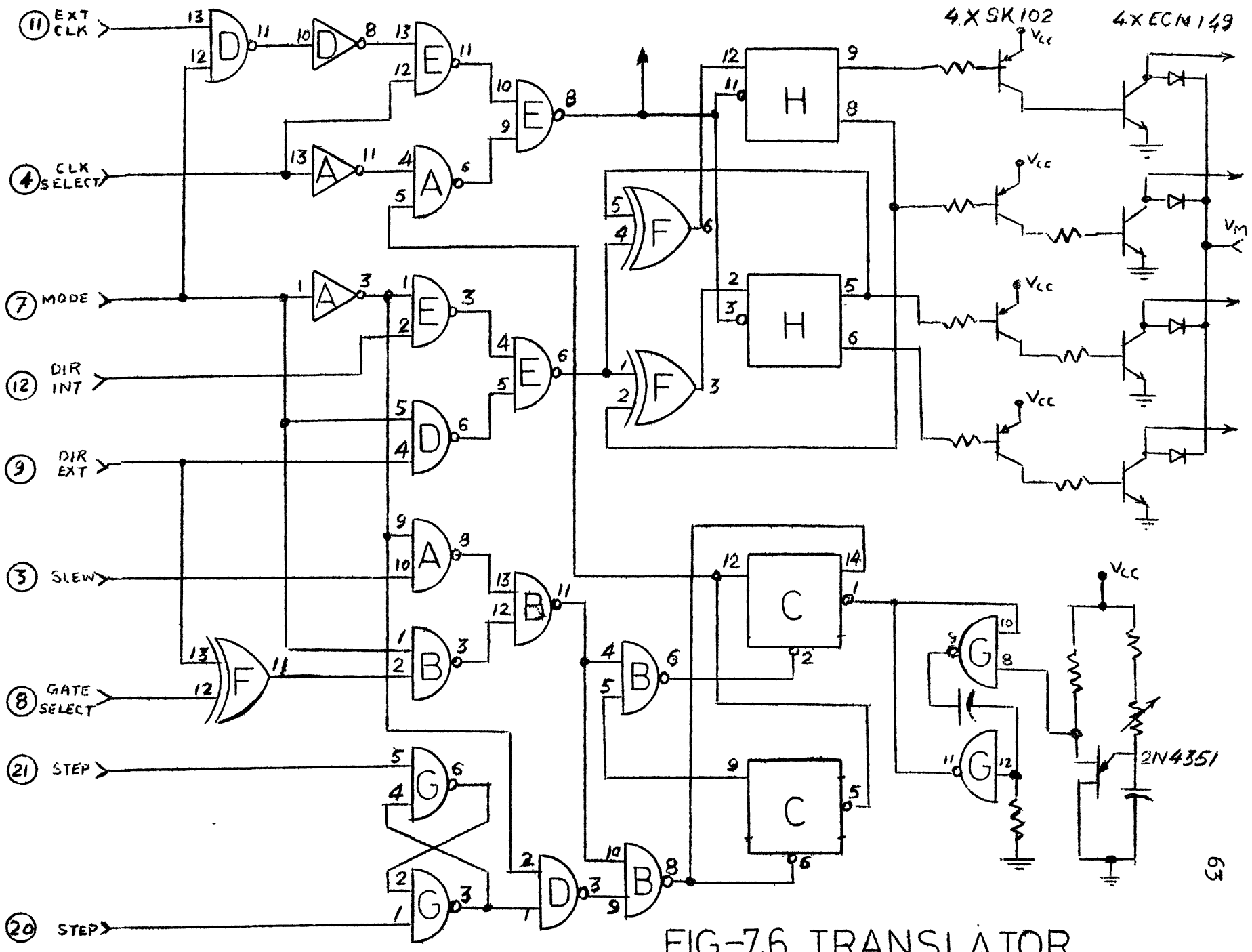


FIG-7.6 TRANSLATOR

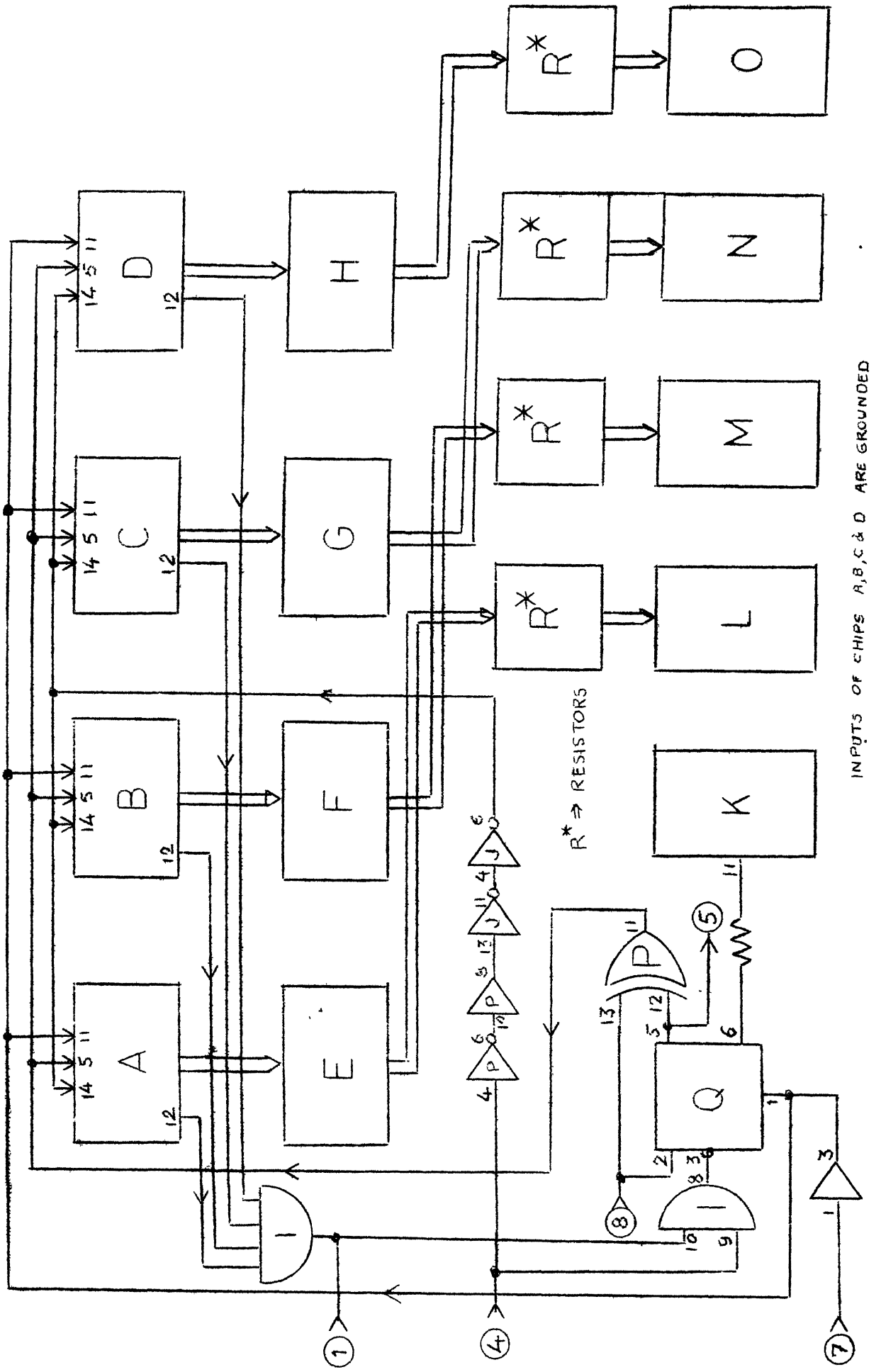


FIG-7.7 POSITION DISPLAY

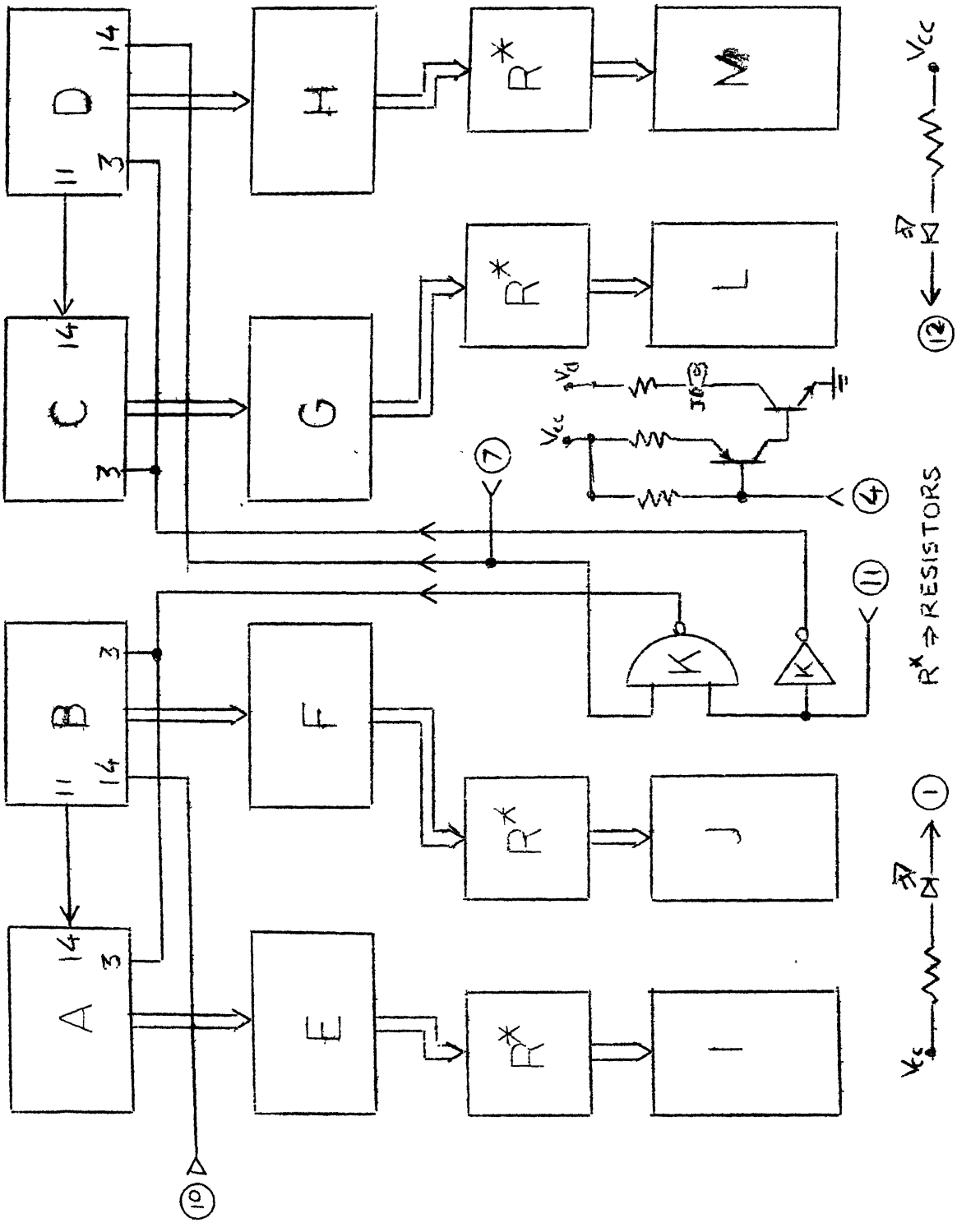


FIG-7.8 PASSES AND CYCLES DISPLAY

Table 7.2: CARD LAYOUTS

CLASS	CARD NO	CHIPS USED		POWER DISSIPATION
SYSTEM	7.1	SN7402-H,M,N SN7495-A,B,C,E,F,G SN74191-I,J,K	SN7421-L SN74121-D	2880 mW
	7.2	SN7402-D SN7495-A,B,C,E,F,G SN74121-I	SN7421-H SN7497-J,K SN74191-L,M,N	3480 mW
CONTROL	7.3	SN7400-S SN7407-F SN7420-M,N SN7476-K	SN7402-H,L SN7408-A,G,J,P SN7473-I,O,Q,R SN74121-B, C,D,E	1670 mW
	7.4	SN7400-B,C,D,F,K SN7408-I SN7420-J SN7486-A	SN7402-E SN7413-G SN7473-H	530 mW
	7.5	SN7400-B,C,D,E,H SN7402-F SN7421-A	SN7401-I SN7420-G	820 mW
TRANS-LATOR	7.6	SN7400-A,B,D,E,G SN7474-H	SN7473-C SN7486-F	505 mW
DISPLAY	7.7	SN7400-J SN7447-E,F,G,H SN7486-P SLA-7 K,L,M,N,O	SN7421-I SN7474-Q SN74190-A,B,C,D	4365 mW
	7.8	SN7400-K SN7490-A,B,C,D	SN7447-E,F,G,H SLA-7 I,J,L,M	3500 mW

TABLE 7.3: CARD PIN-CONNECTIONS

Pin Nos. \ Fig.No.	7.1	7.2	7.3	7.4	7.5	7.5	7.6	7.7	7.8
1	GND	GND	GND	GND	Interpo-lation	V_{CC}	NC	$M_{x/y}$	Inter-polation
2	GND	$\bar{Q}_R$	x + y	GND	Cycle over	V_{CC}	V_{CC}	V_d	V_{CC}
3	Sr.Data (in)	Sr.Data (in)	Sr.Data (out)	POT	MODE	NC	SLEW	V_d	V_{CC}
4	V_{CC}	V_{CC}	Word Clk	Clk Select (x)	JO	NC	Clk Select	Drive Pulses	JO
5	V_{CC}	V_{CC}	V_{CC}	V_{CC}	NC	MODE	NC	S_D	V_d
6	Sr.Clk (in)	Sr.Clk (in)	Sr.Clk (out)	Clock select (y)	DS	Q_S	GND	NC	V_d
7	Mode	Mode	Mode	C_o(out)	NC	NC	Mode	Clear	Cycle over
8	C_o(in)	C_o(in)	C_o(in)	EXT DIR(y)	NC	NC	Gate Select	S_d	GND
9	Feed Clk	Load Pulse	Load Pulse	EXT DIR(y)	NC	Enable Counters	EXT DIR	V_{CC}	GND
10	Load Pulse	S_d	Code Clk	S_d(x)	NC	NC	NC	V_{CC}	Pass over
11	PRM Clk	PRM Pulses	$\bar{Q}_{SYNC}$	S_d(y)	NC	$\bar{Q}_R$	EXT Clk	GND	Clear
12	NC	TP	IE	Clk I_{FF}	NC	Code Clk	DIR INT	GND	Retrace

(Continued)

Table 7.3 Continued.

Pin Nos. \ Fig.No.	7.2	7.2	7.3	7.4		7.5	7.6	7.7	7.8
13	TP	TP	Sr.Data (in)	D_{xy}	-	D_{xy}	Drive pulses	-	-
14	TP	Sr.Data (out)	Sr.Clk (in)	$S_D(x)$	-	Error (IE/OE)	TP	-	-
15	TP	$M_{x/y}$	D_{xy}	$S_D(y)$	-	Q_{SYNC}	NC	-	-
16	TP	NC	$\overline{Q_S}$	$\overline{Q_{JO}}$	-	M_y	NC	-	-
17	Enable Counters	TP	RESET	Q_{IE}	-	M_x	NC	-	-
18	TP	TP	JO	M_x	-	$x \oplus y$	NC	-	-
19	NC	PRM Clk	Demand Clk	M_y	-	Retrace	NC	-	-
20	NC	TP	+	+	-	V_d	STEP	-	-
21	TP	TP	X	X	-	GND	STEP	-	-
22	D_{xy}	TP	Y	Y	-	GND	NC	-	-

(v) Job over indicator light.

(vi) Ready light indicating state of the Ready Latch. System can be started only when this light is ON.

(vii) Running Indication - This LED keeps going ON and OFF during the process.

(viii) INPUT ERROR Light - This light indicates the failure of arrival of data from the SAM in spite of demand sent.

(ix) OUTPUT ERROR Light - This light indicates instantaneous execution error more than the tolerance limit.

7.4 PANEL CONTROL SWITCHES

(i) CLEAR DISPLAY - It clearsthe counters storing x and y positions, passes completed and cycles completed. This is used to define ORIGIN and hence it must not be pressed until a new origin has to be defined.

(ii) MANUAL/AUTO - This switch determines whether CONTROL is to be manual or by a program.

(iii) Forward/Reverse - It is used to determine direction of movement of stepper motor in manual mode.

(iv) STEP - It causes the motor to move by a single step in the selected direction in manual mode.

(v) SLEW - It causes the motor to run continuously in the selected direction as long as it is kept pressed in the manual mode.

(vi) HALT - It halts the movement of the system as soon as it is pressed.

(vii) STOP - It stops the system only when no data is left in the system registers (EREG and BREG).
(viii) RESET - It is used to clear the IE, OE and JOB OVER latches to enable START either when starting the next job or after correcting the error which might have occurred.
(ix) FEED RATE - Used to vary the rate of movement of the table (bed).

7.5 PANEL CONNECTORS

At the Front Panel a connector is provided for connecting it to the Serial Access Memory and at the back panel two connection have been provided for connecting the stepper motors which drive the bed-positioner.

7.6 OPERATING INSTRUCTIONS

7.6.1 1. Ensure that CONTROL and MOTOR enabling switches are OFF.

2. Connect SAM and motors to the system.

3. Switch MODE to MANUAL.

4. Switch ON the AC mains.

5. Switch ON the CONTROL. READY, HALT and STOP will turn ON.

6. Energise the motors by motor switch.

7. Bring the machine table to the point to be defined as ORIGIN by DIRECTION, SLEW and STEP controls provided on the console for both the axes.

8. Push CLEAR to define the reached point as ORIGIN. All the numeric displays will show zero.

7.6.2 NORMAL OPERATION

1. Switch MODE to AUTO.

2. Push START. READY, HALT and STOP indications go OFF.

3. The machine starts moving in accordance with the data fed by SAM and INTERPOLATION indication makes transitions between ON and OFF states only if the INPUT ERROR indication does not come up. Each transition indicate execution of the next data.

7.6.3 CODE FACILITIES

(a) PASS OVER - The RETRACE indication glows when the machine works under the PASS-Code when the retrace to ORIGIN is complete, the PASSES OVER display is incremented by one and new data is automatically fetched to continue the process.

(b) CYCLE OVER - The STOP indication glows when this code is sensed and the machine comes to a stop after the execution of the current data is completed. The READY indication comes up implying that the machine can be re-started by pressing the START push-button. Also, the CYCLES OVER display is incremented by one and the PASSES display is cleared.

(c) JOB OVER - The STOP indication comes and the machine comes to a stop as in CYCLES OVER code, except, that after the execution of the current data is completed, the READY does not light up, instead the JOB OVER indication is lighted. The machine cannot be started by pressing START

push-botton until the system is reset by the RESET push-button, which turns OFF the JOB OVER light and turns ON the READY light.

7.6.4 MANUAL INTERRUPT FACILITY

(a) HALT - On pushing the HALT push-button, the HALT and READY indications are turned ON and the machine stops at once but, if the machine is retracing under the PASS code then the machine does not stop and READY also does not light up even though HALT is glowing. However, the machine can be halted even when retracing under the PASS code simply by switching MODE to MANUAL. Operation is then resumed whenever MODE is switched back to AUTO. If the READY indication does not light up but the machine stops on pressing HALT, it implies that the PASS code has been sensed by the system and after the current data is executed, RETRACE will take place.

(b) STOP - On pushing the STOP push-button, the STOP indication glows and the machine comes to a stop when no relevant data is left in the system. The STOP indication does not glow on pushing the STOP push-button implies that the next data corresponds to PASS code and the STOP control is ignored. If it is essential that the system should not take any more data then the STOP push-button should be kept pressed when the machine is approaching the defined ORIGIN under the PASS code till the STOP and READY indications glow.

7.6.5 ERROR INDICATIONS

A simple trouble-shooting chart is given below which will facilitate the user to make a quick check on the overall system interconnections. In case the error persists then a complete check up of the controller will of-course be necessary.

TABLE 7.4: TROUBLE-SHOOTING CHART

ACTION	OBSERVATION	REMEDY
CONTROL switched ON	Numeric display or HALT or STOP or READY does not glow	Check power supplies
SLEW/STEP push-buttons pressed	Motors do not respond	Check motor power supply
START pushed	READY, HALT, STOP go OFF but the machine does not move.	Check the MODE switch, it should be in AUTO position. If it is in AUTO, it means that the first data supplied is a code which is not permitted. The process can be re-started by switching the CONTROL to OFF and then to ON position.
START pushed	As above and INPUT ERROR light comes up	Check SAM(data is not supplied inspite of Demand being sent).

CHAPTER 8

TEST RESULTS AND CONCLUDING REMARKS

This chapter describes the testing circuit required for the controller followed by the tests made using the test circuit. The chapter is concluded by the description of test made and system response to the test.

8.1 TEST CIRCUIT

As the Numerical Controller fabricated does not have a memory to store the whole of the program, it requires a memory which should supply to it the desired data whenever the need arises. This requirement makes it dependent on another equipment which has earlier been named as SAM, the Serial Access Memory. To be able to test this instrument independently, a simple circuit has been designed and fabricated.

The test-circuit has to supply the serial clock, data and word clock whenever a demand is sent to this circuit by the system. To be able to supply the data for one interpolation interval, a memory of at least 36 bits (three words of 12 bits each) is required. Here, to simplify it further, eight least significant bits of each word have been taken to be zero for ever and instead a gated burst clock is fed to three shift registers (each shift-register corresponding to four most significant bits of each word). The sign and magnitude of the words 'x' and 'y

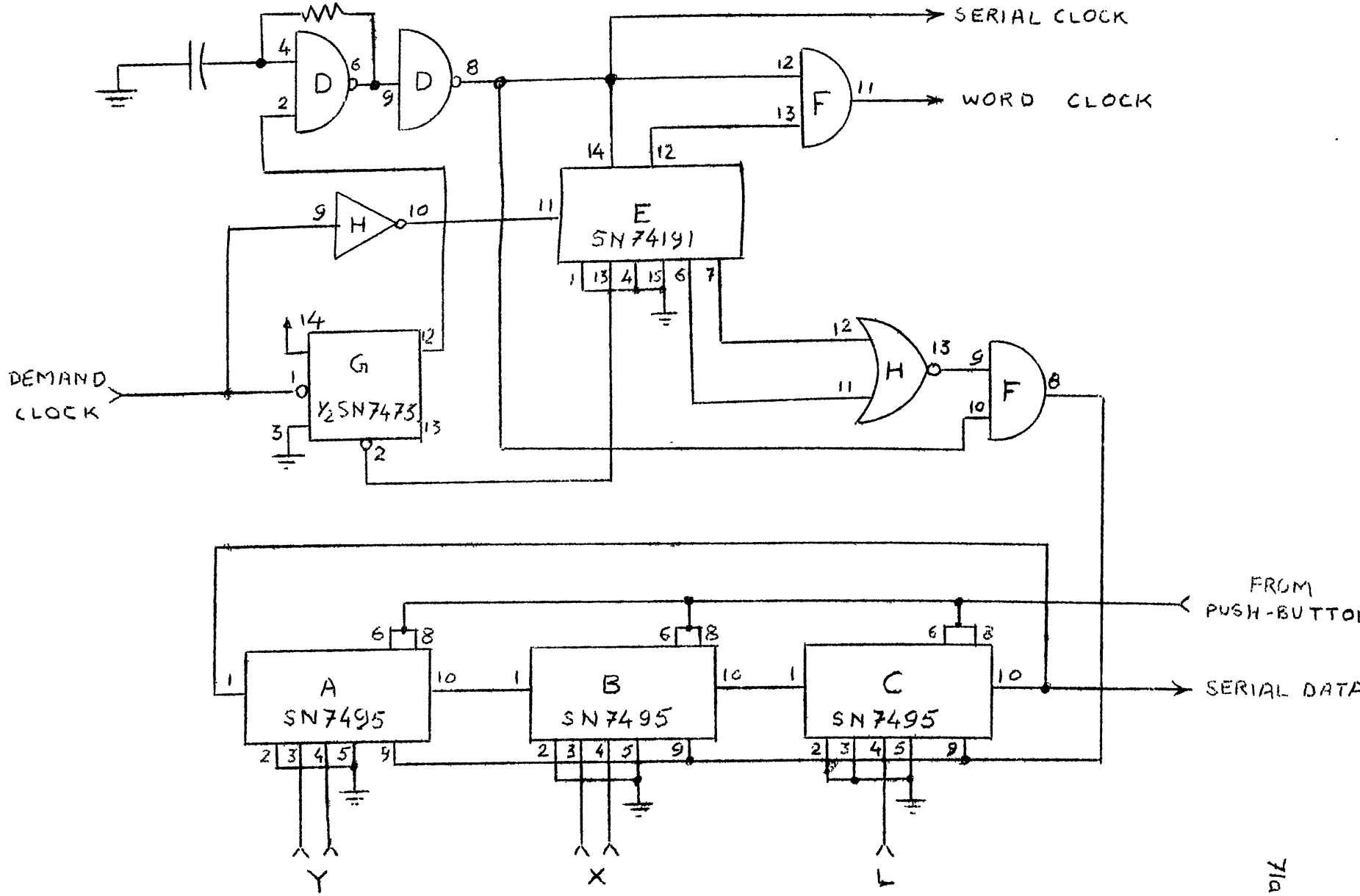


FIG-8.1 TEST CIRCUIT

and magnitude of the word 'L' can be changed by switches provided on the front-panel of the test-box. The data corresponding to these switch positions is fed to the shift-registers by pressing the push-button provided on the front-panel of the test-box. The complete circuit is given in Figure 8.1. The table given below explains the different combinations of data that can be fed using the test-box and their interpretations.

TABLE 8.1 DATA INPUT BY TEST-BOX

M_L	S_X	M_X	S_Y	M_Y	INTERPRETATION
0	0	0	0	0	JOB OVER Code
0	0	0	A*		PASS Code, retrace takes place first along y-axis and then along x-axis.
0	A		0	0	PASS Code, retrace takes place first along x-axis and then along y-axis.
0	A		A		CYCLE OVER Code
1	Ø@	0	Ø	1	Movement along y-axis in positive or negative direction depending upon whether S_Y is zero or one respectively.
1	Ø	1	Ø	0	Movement along x-axis in positive or negative direction depending upon whether S_X is zero or one respectively.
1	Ø	1	Ø	1	Movement along both the axes simultaneously, the slope and direction being dependent on S_X and S_Y.
1	0	0	0	0	Impossible situation, not permitted.

* A — 01, 10, 11

@ Ø — 0 or 1

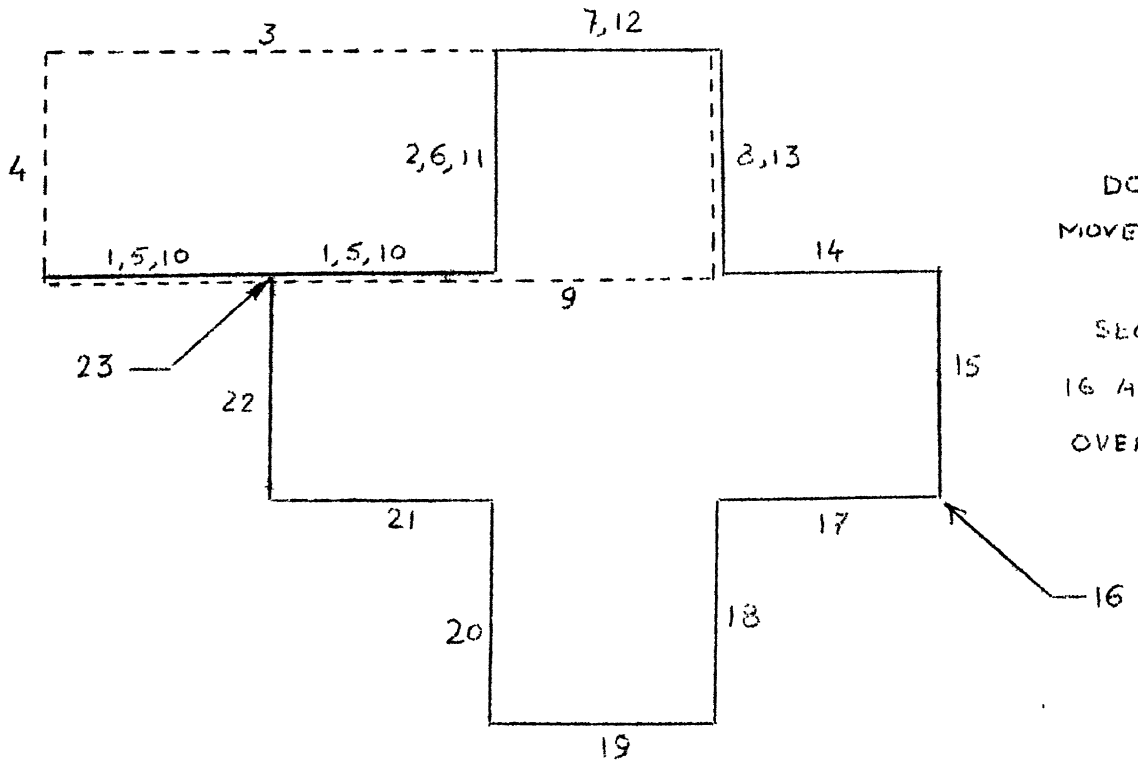
8.2 TEST PROCEDURE

Connect the test-box to the system and do the 'Initial Adjustments' as explained in Section 7.6.1. Set the data by the switches and feed it by pressing the push-button all on the front-panel of the test-box. Follow the instructions for 'Normal Operation' given in Section 7.6.2. Set the next data quickly on the test-box and feed it before the next demand comes. The next demand is indicated by the change in the state of INTERPOLATION LED on the console of the system. If the push-button on the test-box is kept pressed and demand comes, then it will lead to erratic operation of the system.

Various test patterns were tried on the system by different combinations of data fed to the system. One example is illustrated here for which the test-pattern is given in Figure 8.2. Different segments of the test-pattern have been named which are traced out by the data fed, which is given in the table below.

TABLE 8.2 TEST PATTERN TABLE

Segment No.	L	X	Y	Segment No.	L	X	Y
1	1	0 1 0 0		13	1	0 0	1 1
2	1	0 0 0 1		14	1	0 1	1 0
3, 4	0	A	0 0	15	1	0 0	1 1
5	1	0 1 0 0		16	0	A	A
6	1	0 0 0 1		17	1	1 1	1 0
7	1	0 1 0 0		18	1	1 0	1 1
8,9	0	0 0 A		19	1	1 1	1 0
10	1	0 1 0 0		20	1	1 0	0 1
11	1	0 0 0 1		21	1	1 1	0 0
12	1	0 1 0 0		22	1	1 0	0 1
				23	0	0 0	0 0



DOTTED LINES INDICATE MOVEMENT UNDER PASS CODE

SEGMENTS (POINTS MARKED 16 AND 23) REPRESENT CYCLE OVER AND JOB OVER CODE RESPECTIVELY

FIG-8.2 TEST PATTERN

BIBLIOGRAPHY

1. A.A. Shumsheruddin, 'A New Generation of Continuous Path N/C Systems', Advances in Machine Tool Design and Research pp.1191-1207, 1968 (Part-2).

2. V.V. Karibskii, 'On the Error of a Linear Interpolation for a Program Controlled Digital System', Autom tion and Remote Control, Vol.20, pp. 725-32, Jan-Sept. 1959.

3. Z. Yang, 'Determination of Maximum Error of a Binary Multiplier', Automation and Remote Control-21, No.7, pp. 703-713, 1961.

4. R. Bell, A.C. Lowth and R.B.Shelley, 'The Use of Stepper Motors in Numerically Controlled Machine Tools', International Journal of Machine Tool Design and Research, Vol.10, No.4, pp. 417, 1970.

5. S.J.Bailey, 'Incremental Servos', Control Engg., Vol.7, No.11, Nov. 1960-May 1961.

6. D.F. Walker, 'Data Processing for N/C of Machine Tools', The Radio and Electronics Engineering -26, pp. 501, 1963.

7. T.T. Smith, 'Application of Stepping Motors in Machine Tool Systems', Symposium on Incremental Motion Control System 1, Urbana-Champaign, Illinois, pp. 587-604, pp. 27-31, March 1972.

8. J.T. Evans and L.U.C. Kelling, 'Inside the Mark Century N/C', Control Engg., Vol.10, No.5, pp. 112, May 1963.

9. H.B.Benegar and T.L.Zinn, 'New Continuous Path System uses D.D.A. Interpolator', Control Engg., Vol.8, No.1, pp. 71, January 1961.

10. J.D. Martin, 'Signal Processing and Computation Using Pulse Rate Technique', Radio Electronics Engg., No.38, pp. 329, 1969.

11. Vyngvar Lundh, 'Digital Techniques for Small Computations', Journal of British Institute of Radio Engineering, Vol.19, pp. 37, Jan. 1959.

12. R. Phillips, 'Many Digital Functions can be Generated with Rate Multiplier', Electronic Design, Vol.16, No.3, pp.82, 1968.

13. R. Edwards, 'Pulse Rate and Pulse Number Signals in Automatic Control', Control, Vol.12, pp. 627-630, July, 1968.

14. R.N.Nicola, 'Operational Digital Techniques', Aeronautical Engg. Review, Vol.15, pp. 78-82, March 1956.

15. J. Plas, J. Blommaert, ''A Stepping Motor Drive Assembly especially Designed for CNC Systems', 13th International N/C Tool Design Research Conference, Birmingham, London, paper 44/p-8, Sept., 1972.

16. G. Garratt, 'NC Machine Tools under Direct Computer Control', Mach.Prod. Engg. (GB), Vol.115, No.2955, p. 2, July 1969.

17. G.Widl, 'Possibilities and Methods of Numerical Contour Control', Electrotech. Z (ETZ)B, Vol.21, No.12, pp.276-284, June 13, 1969.

18. R.Bell, A.C.Lowth and R.B.Shelley, Application of Stepping Motors to Machine Tools. The Machinery Publishing Company Ltd., London, 1970.

19. Simon Ing Wilhelm, Numerical Control of Machine Tools. Edward Anold (Publishers) Ltd., London, 1973.

20. G.G.Ertell, Numerical Control, Wiley Interscience, New York, 1969.

21. Childs, Numerical Control Part-Programming. Industrial Press Inc., New York, 1973.

CARD INTER-CONNECTIONS TABLE

	A	B	C	D	E_1	E_2	F	G	H	I					
1	GND	GND	GND	GND	GND	K-1 Interpolation	V_{cc}	NC	NC	E_1-18 M_x	8-15	E_1-19 M_y	8-15	E_2-1 Interpolation	
2															
2	GND	GND	GND	F-11 $\overline{Q}_R$	F-18 $x \oplus y$ (retrace)	K-7 Cycle over	V_{cc}	V_{cc}	V_{cc}	V_d		V_d		V_{cc}	
3	B-14 Sr.Data (in)	C-14 Sr.Data (in)	D-3 Sr.Data (in)	C-3 Sr.Data (out)	Pot.(from console)	Mode(from console)	NC	Slew(from console)	Slew(from console)	V_d		V_d		V_{cc}	
4	V_{cc}	V_{cc}	V_{cc}	Word Clk. (from SAM)	G-4 Clk. Select(x)	K-4 JO Signal	NC	E_1-4 Clk. Select	E_1-6 Clk. Select	G-13 Translator pulses		H-13 Translator pulses		E_2-4 JO Signal	
5	V_{cc}	V_{cc}	V_{cc}	V_{cc}	V_{cc}	NC	Mode(from console)	NC	NC	E_1-14 S_D(x)		E_1-15 S_D(y)		V_d	
6	Sr.Clk. (in)	Sr.Clk. (in)	Sr.Clk. (in)	Sr.Clk. (out)	H-4 Clk.Select (y)	F-14 DS	D-16 Q_S	GND	GND	NC		NC		V_d	
7	D-7 Mode (in)	D-7 Mode (in)	D-7 Mode (in)	Mode (out)	D-8 C_o(out)	NC	NC	Mode(from console)	Mode(from console)	Clear(from console)		Clear(from console)		E_2-2 Cycle over	
8	A-9 C_o (in)	C_o (in)	C_o (in)	E_1-7 C_o (in)	G-9 EXT DIR (x)	NC	NC	E_1-8 Gate Select	E_1-9 Gate Select	E_2-10 S_d		E_2-11 S_d		GND	
9	A-8 Feed Clk.	C-9 Load pulse (in)	D-9 Load pulse (in)	E_1-12 Load pulse (out)	H-9 EXT DIR (y)	NC	A-17 Enable counters	E_1-8 EXT DIR	E_1-9 EXT DIR	V_{cc}		V_{cc}		GND	
10	C-9 Load pulse (in)	E_1-10 S_d(x)	E_1-11 S_d(y)	F-12 Code Clk. (out)	B-10 S_d(x)	NC	K-10 Pass over	NC	NC	V_{cc}		V_{cc}		F-10 Pass over	
11	B-19 C-19 PRM Clk. (out)	G-11 Drive pulses	H-11 Drive pulses	F-15 Q_{sync}	C-10 S_d(y)	NC	D-2 $\overline{Q}_R$	B-11 EXT Clk.	C-11 EXT Clk.	GND		GND		Clear (from console)	
12	NC	TP	TP	$\overline{Q}_{IE}$(to console)	D-9 Clk.I_{FF}	NC	D-10 Code Clk. (in)	DIR INT (from console)	DIR INT (from console)	GND		GND		F-19 Retrace	
13	TP	TP	E-13 D_{xy}	Sr.Data (from SAM)	C-13 D_{xy}(in)	-	C-13 D_{xy}	I-4 Drive pulses	J-4 Drive pulses	-		-		-	
14	TP	A-3 Sr.Data (out)	B-3 Sr.Data (out)	Sr.Clk. (from SAM)	I-5 S_D(x)	-	E_2-6 Error (IE/OE)	TP	TP	-		-		-	
15	TP	I-1 M_x	J-1 M_y	E_1-20 D'_{xy}	J-5 S_D(y)	-	D-11 Q_{sync}	NC	NC	-		-		-	
16	TP	NC	NC	F-6 $\overline{Q}_S$	D-18 $\overline{Q}_{JO}$	-	I-1 M_y	NC	NC	-		-		-	
17	F-9 Enable counters	TP	TP	RESET(from console)	Q_{IE}(to console)	-	J-1 M_x	NC	NC	-		-		-	
18	TP	TP	TP	E_1-16 $\overline{Q}_{JO}$	I-1 M_x	-	E_1-2 $x \oplus y$	NC	NC	-		-		-	
19	NC	A-11 PRM Clk. (in)	A-11 PRM Clk. (in)	Demand Clk.(to SAM)	J-1 M_y	-	K-12 Retrace	NC	NC	-		-		-	
20	NC	TP	TP	TP($\overline{L}$)	D-15 D_{xy}(out)	-	V_d	STEP(from console)	STEP(from console)	-		-		-	
21	TP	TP	TP	E_1-21 X	D-21 X	-	GND	STEP(from console)	STEP(from console)	-		-		-	
22	D_{xy}	TP	TP	E_1-22 Y	D-22 Y	-	GND	NC	NC	-		-		-	